PATENT
Attorney Docket No. 26625.705

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application)	
)	Confirmation No.: 6637
Inventors: Michael T. Loos et al.)	
)	Art Unit: 2154
Application No.: 09/848,769)	
)	Examiner: Chang, Jungwon
Filed: May 3, 2001)	
)	Customer No. 021971
Title: System and Method for Communicating In a Mobile Domain Across Non-Persistent Data Links)	

DECLARATION OF MICHAEL T. LOOS SUBMITTED UNDER 37 CFR 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Examiner Chang:

I, Michael T. Loos hereby declare that:

1. I am one of the inventors of the invention claimed in the above-referenced application. I have read the application and am familiar with its contents.

2. As one of the inventors, I have direct, personal knowledge that the inventors of record in the application conceived the claimed invention and reduced it to practice in the United States. Further, I have direct, personal knowledge that conception and reduction to practice of the claimed invention occurred before April 17, 2000, the earliest date to which the reference cited by the Examiner, U.S. Patent No. 6,636,873 issued to Carini net al., claims priority.

//

3. Attached hereto as Exhibit 1 is a selection of engineering notebook pages dating from January 1999 to June 1999, signed by me, and evidencing the early conception of the invention as well as the continuing efforts of its reduction to practice. Exhibit 1 and a number of technical manuals form U.S. Provisional Patent Application No. 60/202,351, filed May 5, 2000, the provisional application from which the present application claims priority. These pages evidence the Applicants' early conception and reduction to practice of the claimed invention of the present application.

4. Attached hereto as Exhibit 2 is one of a number of technical documents included in U.S. Provisional Patent Application No. 60/202.351, filed May 5, 2000, and to which the present utility application claims priority. Included in Exhibit 2 at page 450 is a revision listing showing that the contents of Exhibit 2 were developed prior to April 17, 2000, the earliest date to which Carini et al. may claim priority. The pages of Exhibit 2 further evidence Applicants' the early conception and reduction to practice of the claimed invention of the present application.

5. For additional support of Applicants' conception and reduction to practice of the claimed invention prior to April 17, 2000, the Examiner may refer to at least the following pages of U.S. Provisional Patent Application No. 60/202,351: pages 232-247 dated January 1999, pages 432-447 dated April 1999, pages 300-319 and 372-431 date May 1999, pages 264-299 dated June 1999, pages 320-371 and 248-263 dated July 1999, pages 142-174 dated September 1999 and pages 178-203 dated November 1999.

6. I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and, further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issuing thereon.

Executed: 5/3/05

By: Michael T. Loos

06 - Jan - 1999

- Omid finishing up first version of AttAdmin interface. More work for tomorrow, including potentially first test of distributed domains

- Ray starting learning process on CE development. Spent long time trying to get CE Services to work between NT and CE machines. Trick ended up being a reinstall of SP3 or SP4 after install of CE Services

- Discussed with Omid the idea of using I/O completion ports for Messaging service, as well as having a separate Messaging & Foundation service. Both seem like favorable ideas at this point.

- Gave Omid the first draft of spec for AttTranslator. He will likely start on that tomorrow or the next day.

- Finished updates to AttDirectory to handle collection types and the concept of collection masters, which are objects that support "parent" collections which are supersets of the collection in question. New objects to the collection can only come from the collection of its collection master.

1-6-99

13-Jan-1999 (Tuesday)

- David has finished first version of AHAdminSrvr. Now working on AHTranslator

- I started review today of AHAdmin — added better error handling — improved Register Domain dialog, including checks for existence of hosts/servers specified by the user

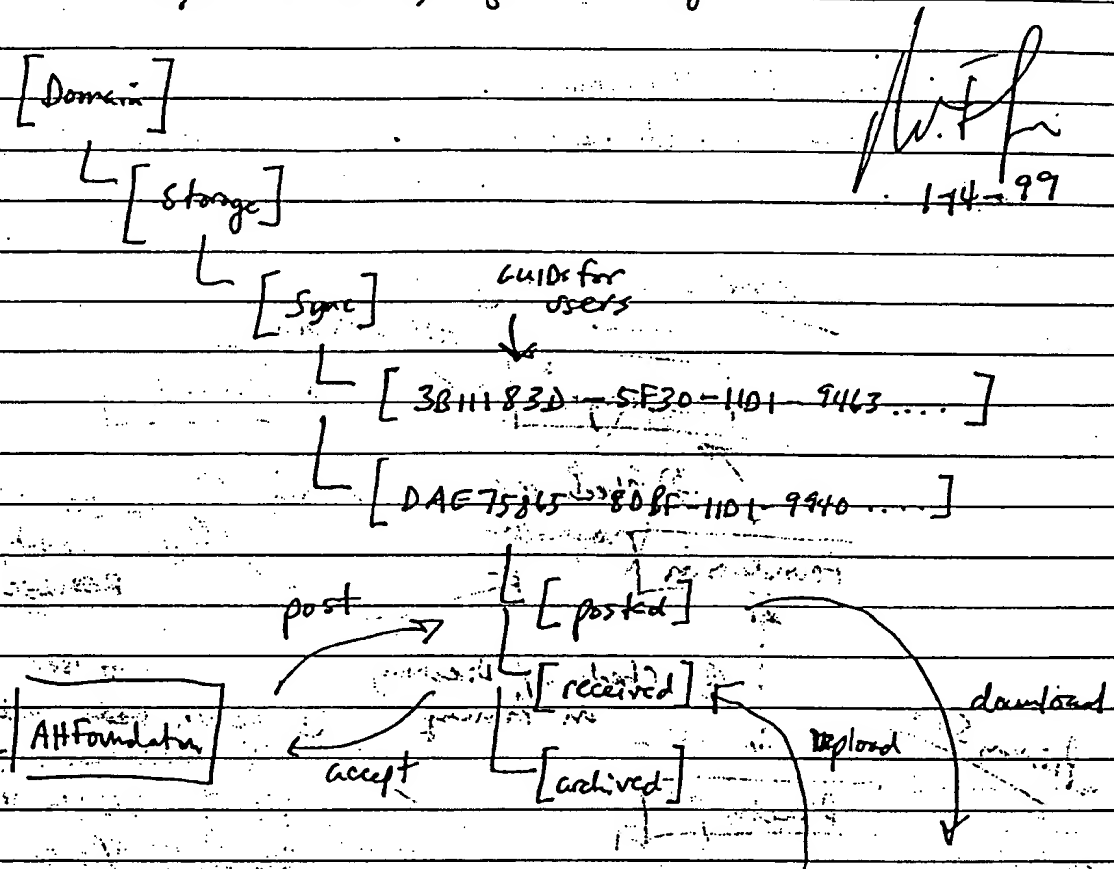
- Yesterday (Monday) got first version of Messenger running on CONCEPTION with Listen Thread, Worker Threads and I/O Completion port. - Built command-line test application to act as client — Made socket connection — passed string — server received it — reversed it and sent back

- Yesterday (Monday) posted Version 0.9 of Foundation spec, including AHFoundation, AHMessenger, and AHLibrary diagrams

1-13-99

14-Jan-1999 (wednesday)

- initial thoughts on workings of AH Library



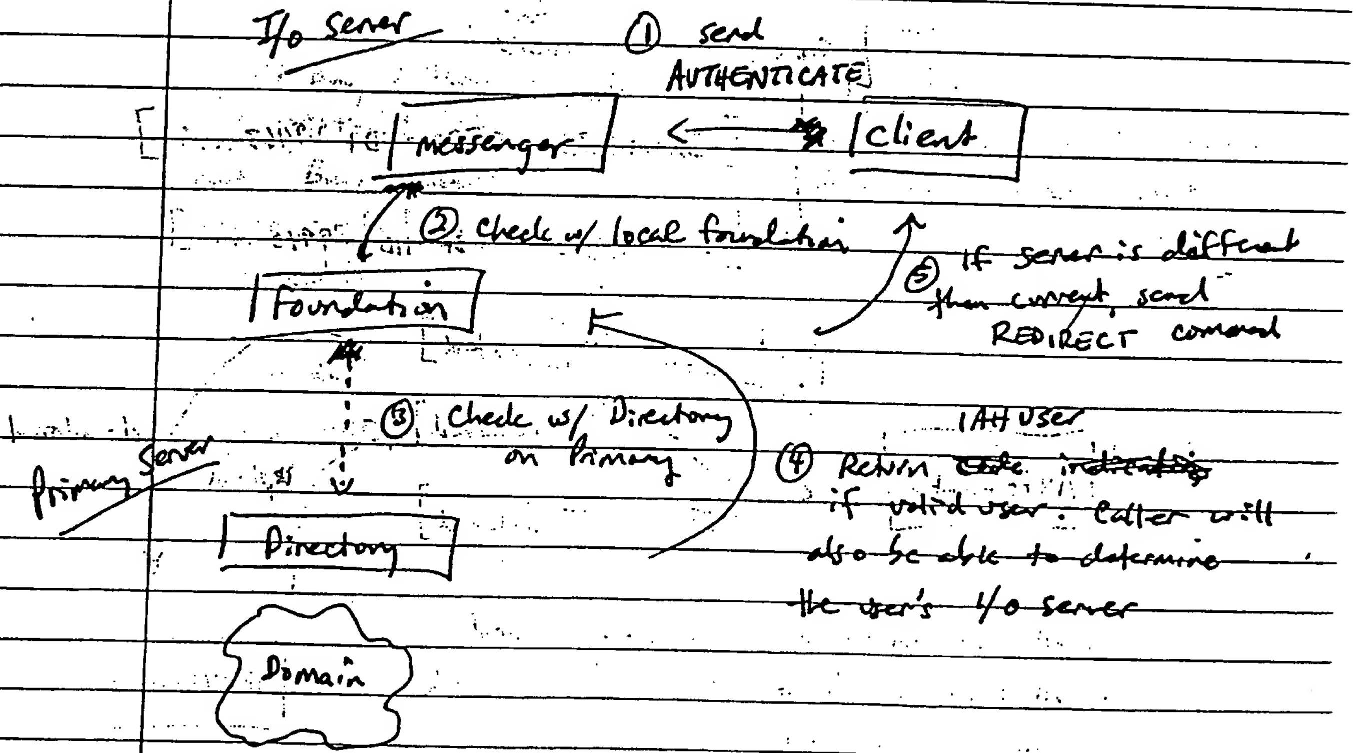
- all files posted or uploaded are archived
- all files are transferred in or out through an IStream interface
- files posted to library are packaged into .MCF files for download (using AH Translator)
- files accepted are unpackaged (AH Translator)

(9:00 AM approx) - meeting with Ray and Omid discussing operation of AH Library in overall synchronization process - Discussed changes Omid made to AH Translator to support inline compression

18-Jan-1999

Connectivity Issues

- REDIRECT command for moving between I/O servers
- variable file chunk sizes based on communication speed


I/O Server
① send AUTHENTICATE
Messenger
Client
② Check w/ local foundation
Foundation
③ Check w/ Directory on Primary
Primary Server
Directory
Domain
④ Return ~~code~~ ~~indicating~~ IAH user if valid user. Caller will also be able to determine the user's I/O server
⑤ If server is different than current, send REDIRECT command

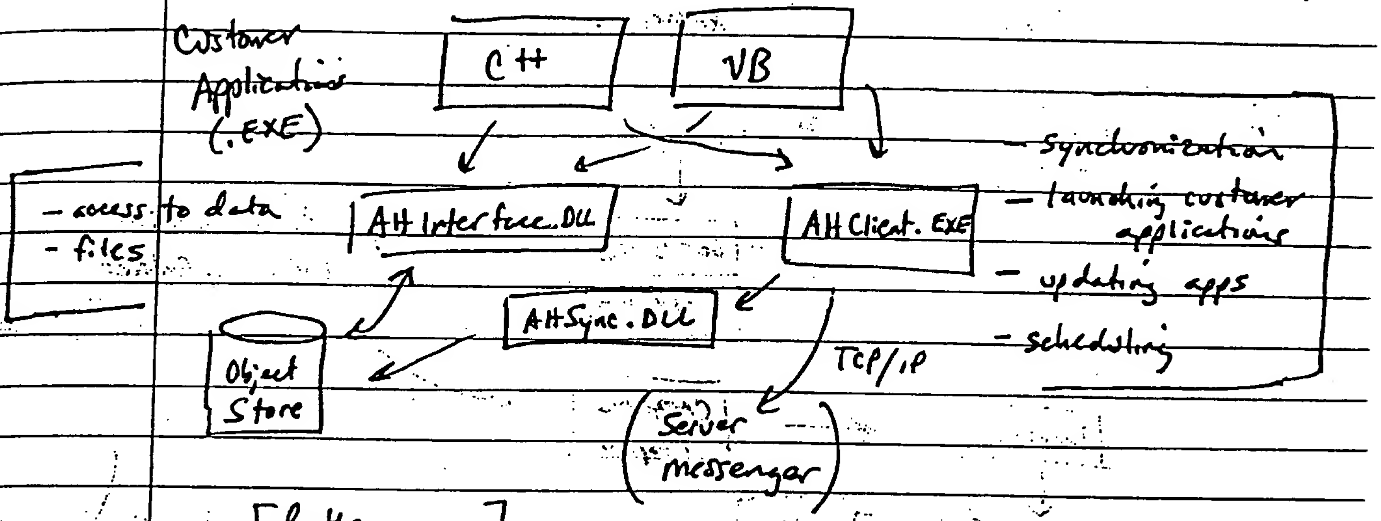
- Messenger will run on all I/O servers and on Primary.
- Messenger will redirect all communication attempts to Primary unless the Primary is the user's I/O server
- ~~All users by default start out with Primary as I/O~~

1-18-99

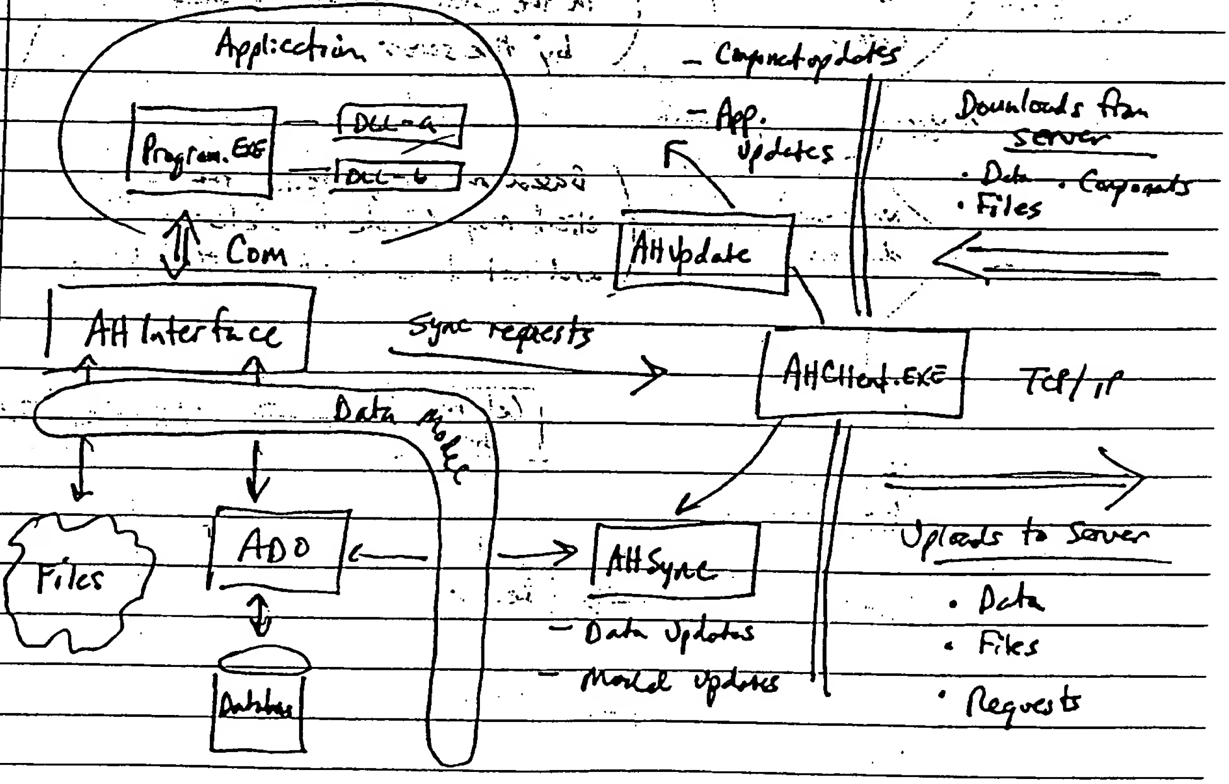
20 - Jan - 1999

— Magellan CE Client

HKEY_LOCAL_MACHINE \ Init
Launch XX = "MyShell.EXE"
Depends XX = hex:14,00



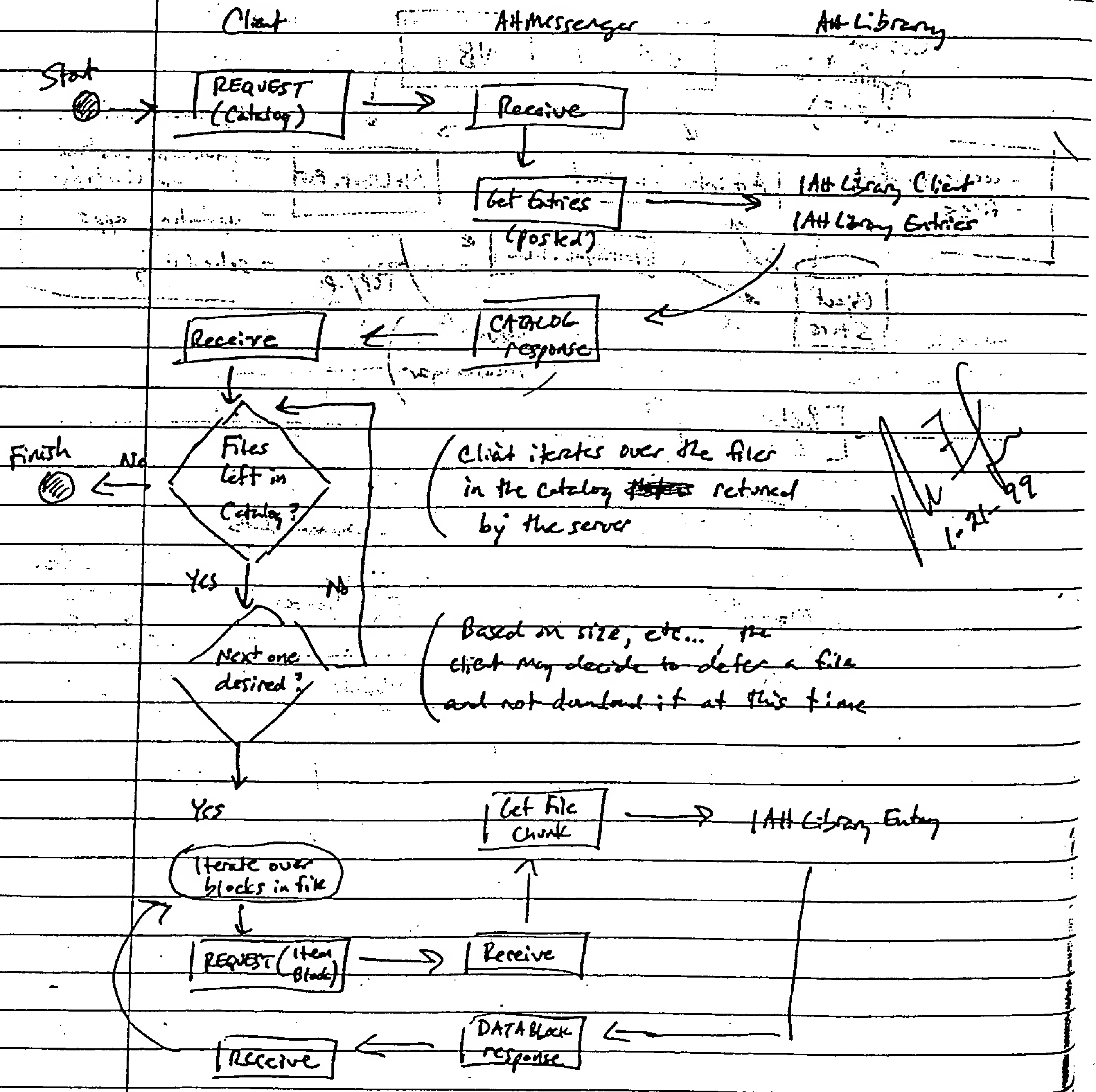
[Better Diagram]



1-20-99

21-Jan-1999

- working on REQUEST and POST commands for Magellan Protocol



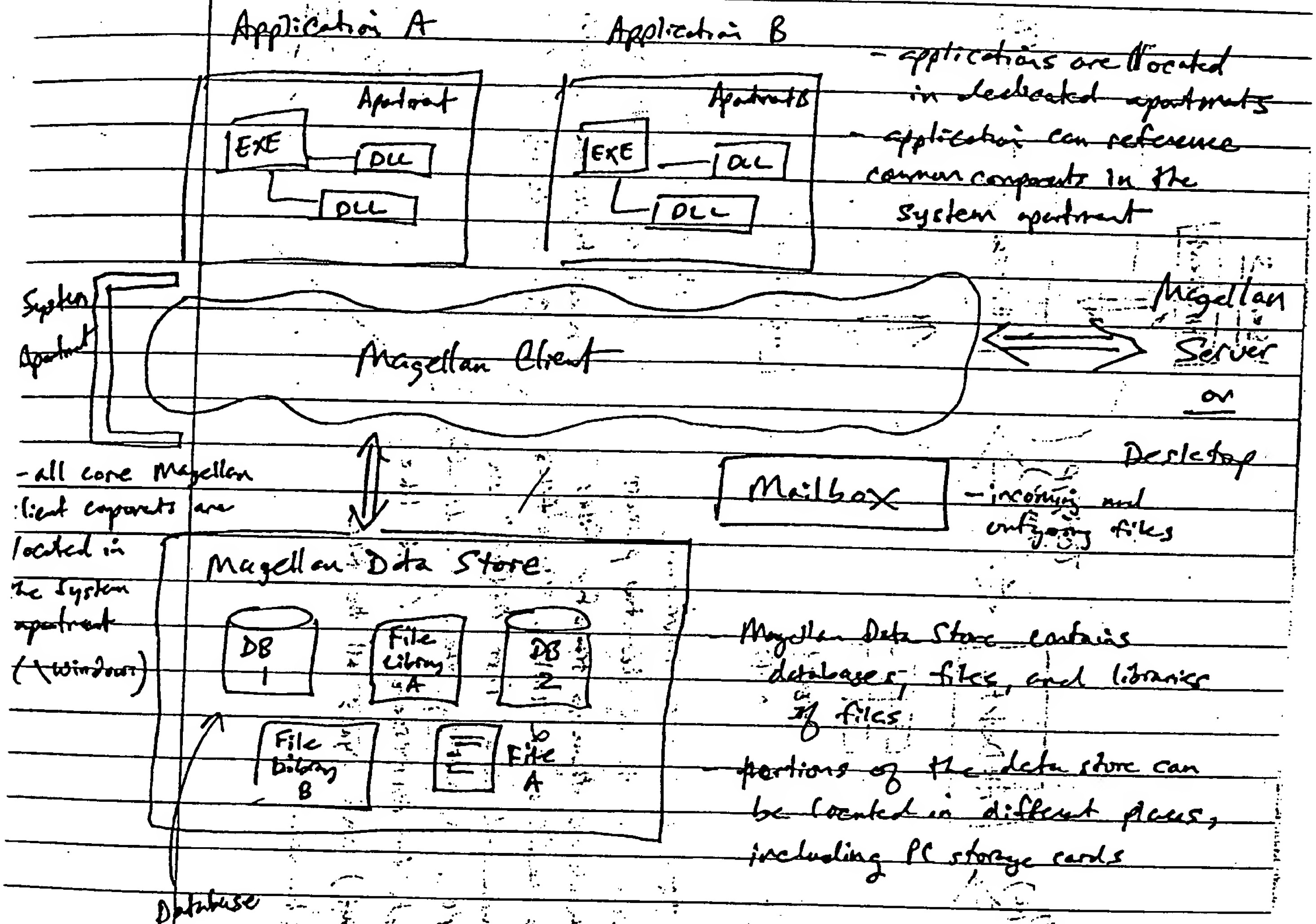
(Client iterates over the files in the catalog ~~[illegible]~~ returned by the server

(Based on size, etc... the client may decide to defer a file and not download it at this time

1-21-99

22-Jan-1999

- specifying high-level client structure



- applications are located in dedicated apartments
- application can reference common components in the system apartment

- all core Magellan client components are located in the system apartment (+ windows)

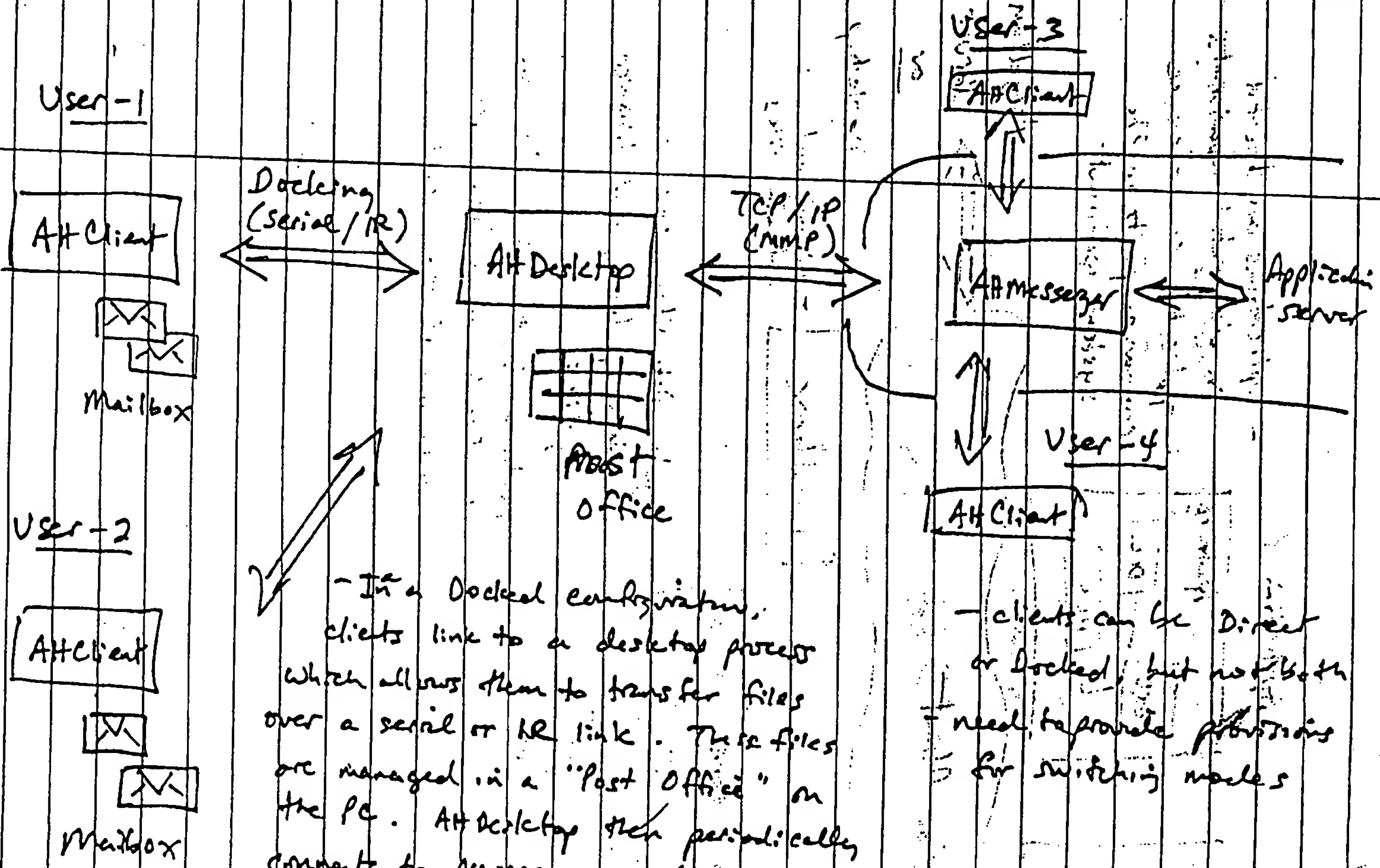
- incoming and outgoing files

- Magellan Data Store contains databases, files, and libraries of files

- portions of the data store can be located in different places, including PC storage cards

1-22-99

23-Jan-1999 (Saturday)



- In a Docked configuration, clients link to a desktop process which allows them to transfer files over a serial or IR link. These files are managed in a "Post Office" on the PC. AH Desktop then periodically connects to AH Messenger (just like a client) and syncs all the files up to the server.

- clients can be Direct or Docked, but not both
- need to provide provisions for switching modes

1-23-99

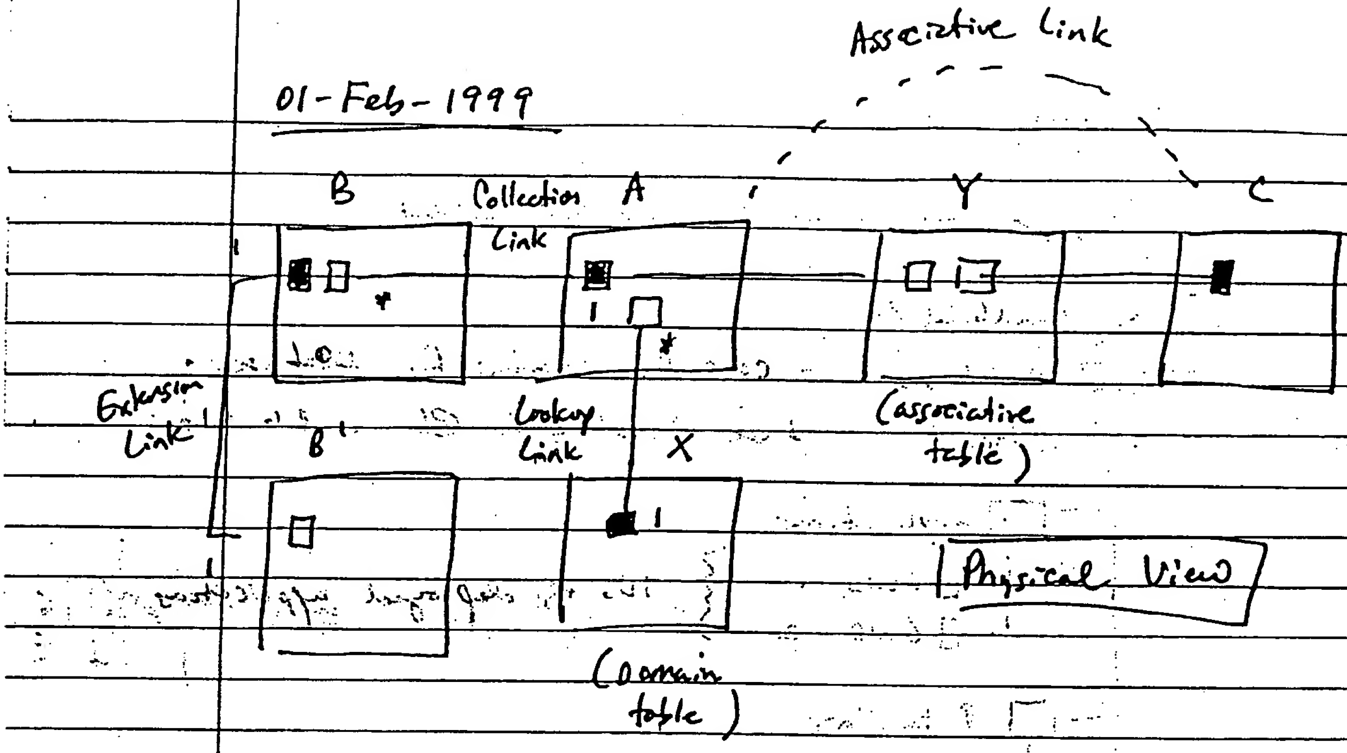
24-Jan-1999 (Sunday)

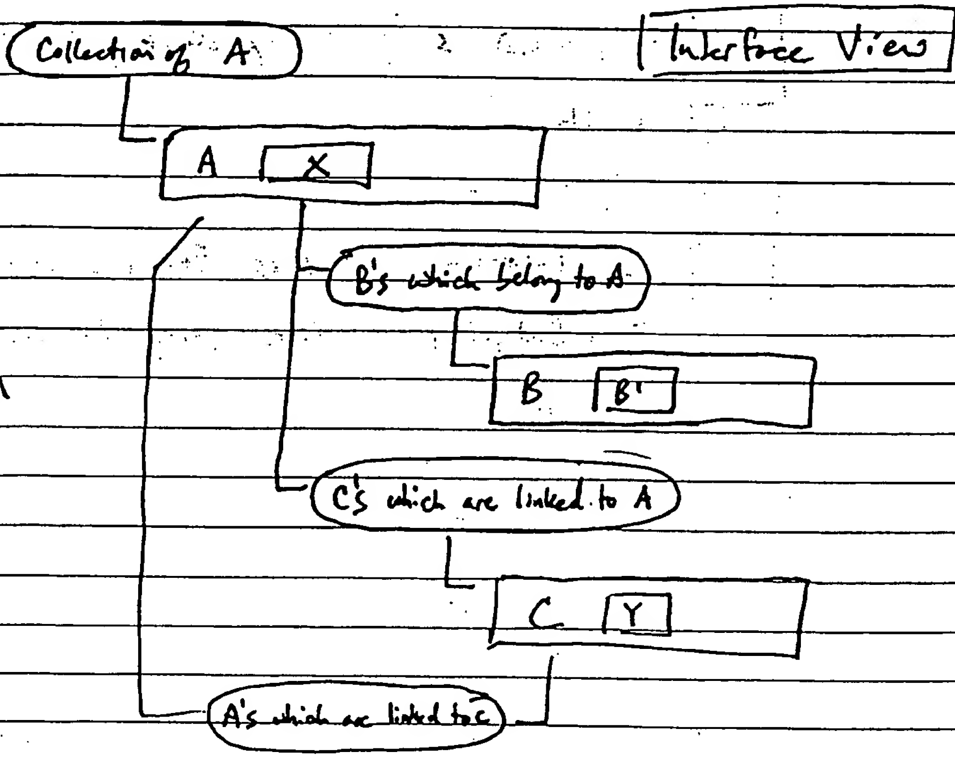
- working on folder structure for CE Client

- □ My Handheld PC
  - □ Databases - core OS location for databases
  - □ @hand → base directory for @hand data files
    - □ Applications
      - □ GUID-1 } IDs of deployed applications
      - □ GUID-2
    - □ Data Store
      - □ Libraries
        - □ GUID-3 } ID of deployed library
      - □ Mailbox
        - □ Archive
        - □ Incoming - files FROM server
        - □ Outgoing - files going TO server
      - □ Submit
    - □ Logs - error log files
    - □ System - data model & configuration files

1-24-99

01-Feb-1999





2-1-99

07-Feb-1999

Data Classes

Magellan Record — Standard, Child, Extension, Lookup, Associative

Magellan File

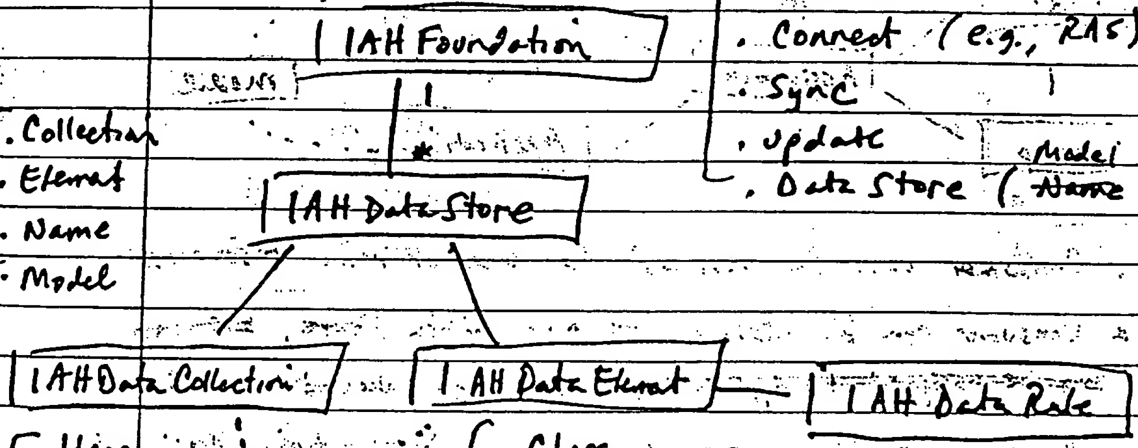
Magellan Folder

Magellan Attribute

Magellan Message

2-7-99

Foundation Interface



IAH Foundation:
- . open
- . Get/Set
- . Connect (e.g., RAS)
- . Sync
- . update
- . Data Store (Model ~~Name~~)

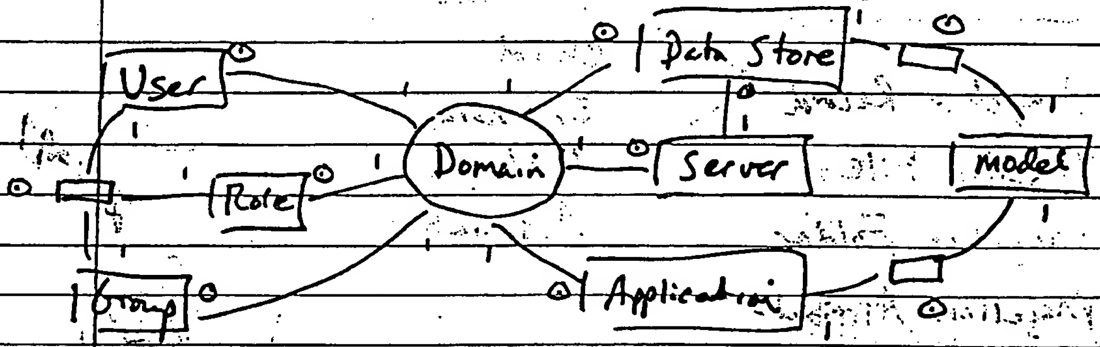
IAH Data Store:
- . Collection
- . Element
- . Name
- . Model

IAH Data Collection:
- . Item
- . Count
- . Class
- . Filter
- . Refresh
- . Add
- . Remove
- . Parent

IAH Data Element:
- . Class
- . Key
- . Field
- . Create
- . Update
- . Delete
- . Rules

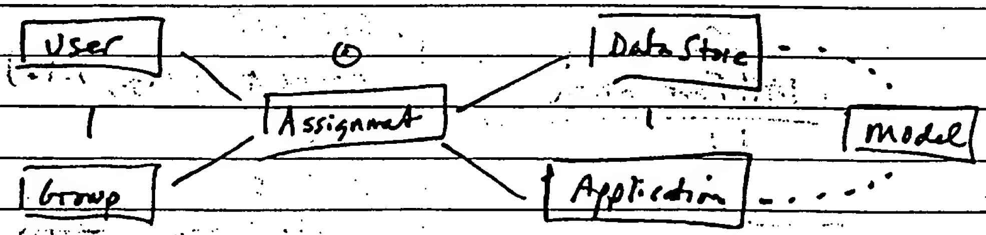
IAH Data Rule:
- . Class
- . Element Key
- . Element Class
- . Receiver Key
- . Receiver Class (e.g., User, Group, Role)

08-Feb-1999 (Monday)



Consumer

Resource

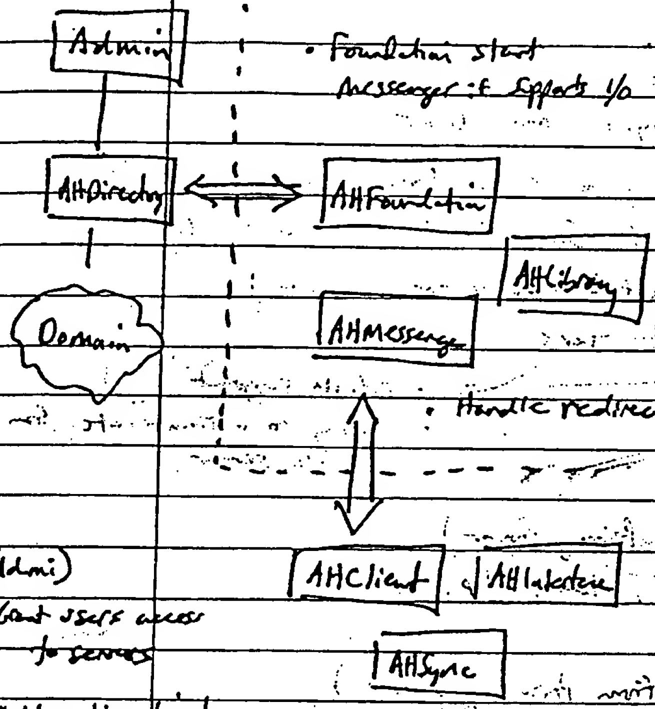


- a Consumer can be assigned to one or more resources
- a Consumer can be assigned to any Data Store, ~~but can only be assigned to Applications which are linked to Models~~
- a Consumer can be assigned to any Application, provided that the Consumer is also assigned to Data Stores which support models that the application may require

2-8-99

2-9-99

09-Feb-1999 (Tuesday)



- Foundation start Messenger if Supports I/O
- Handle redirects

(Admin)

- Grant users access to servers
- Flatten the directory object model

AHRio (AHSubmit AH Retrieve Test Translator)

1) enum users
2) pack files → MTC
3) post files
4) Receive files
5) unpack

AHRio CE (VBSync)

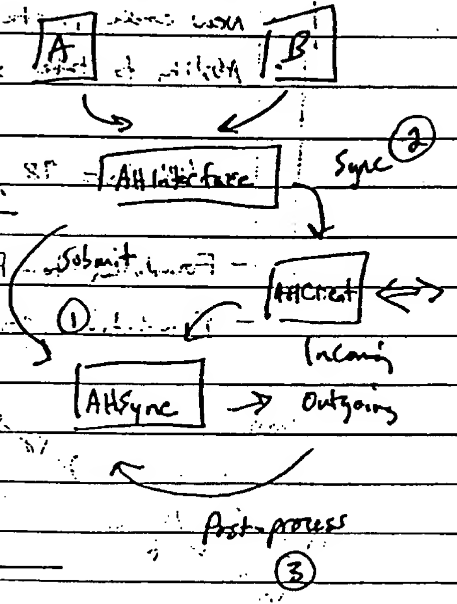
1) Submit
2) Sync.
3) process

Pre-Process (AHSync)

1) Submit entries
2) Files → MTC

Process (AHClient)

1) Sync (called from application)
2) Post
3) Request All
4) Cleanup outgoing
5) Transfer incoming to data store as MTC



Post-Process (AHSync)

1) Explode MTC to native files (selectively)

12-Feb-1999 (Friday)

Foundation Issues

- Messenger performance
- Foundation notification after sync from client
- Foundation file transfer (MTC) to Data Store server
- User movements from Server A to Server B (causing redirect)
- Foundation maintenance of installation (for its server)
- Authentication of "true" domain users
    - Admin support
    - Ability to authenticate from t/o server
- Att library cataloging ?

- BOM for server installation (from Primary server)
- Service installation, de-installation, start, stop, etc.
- Maintenance of MTS components
- New server code deployment (from Primary server)
- New code installation on Primary Server
- Ability to take server "offline" for upgrade

- Windows 98 / 95 / NT version of AttClient (for Desktop)

- Foundation-to-Foundation message transfer
- Foundation connection to Att Directory (

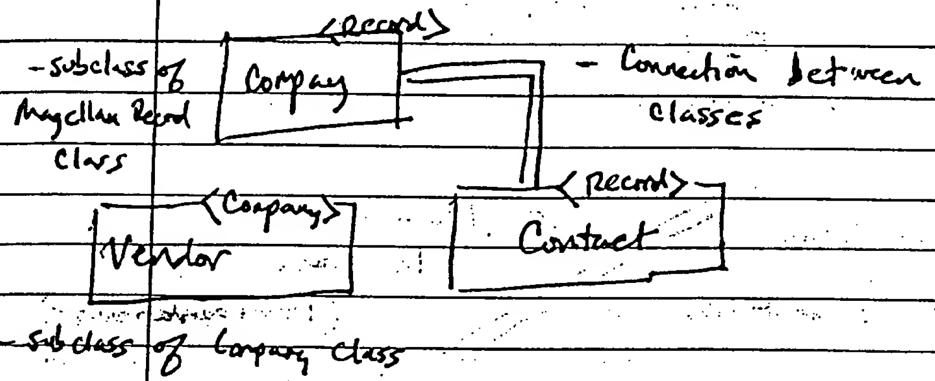
2-12-99

2-20-99

20-Feb-1999 (Saturday)

– Meeting to discuss Developer issues for Milestone B

– Mike L. & Ray Hsu


<Record>
Company
– subclass of Magellan Record class
– Connection between classes
<Record>
Contact
<Company>
Vendor
– subclass of Company class

Dependencies

- Model format (and interface) needed ASAP for Data Engine
- Prototype of Modeler ASAPs for peer & customer review

Priorities

General   Modeler   App Manager   Code Generator

1) XML Format

1a) Interface Component

2) Basic modeler that can be launched from VC & VB

3) Table / List driven interface – double click to properties dialogs

4) Model deployment from developer PC to server and from server to handheld

5) Base code generator from classes (ATL) including logical model algorithms

6) Graphical interface (just viewer)

7) More code generation options

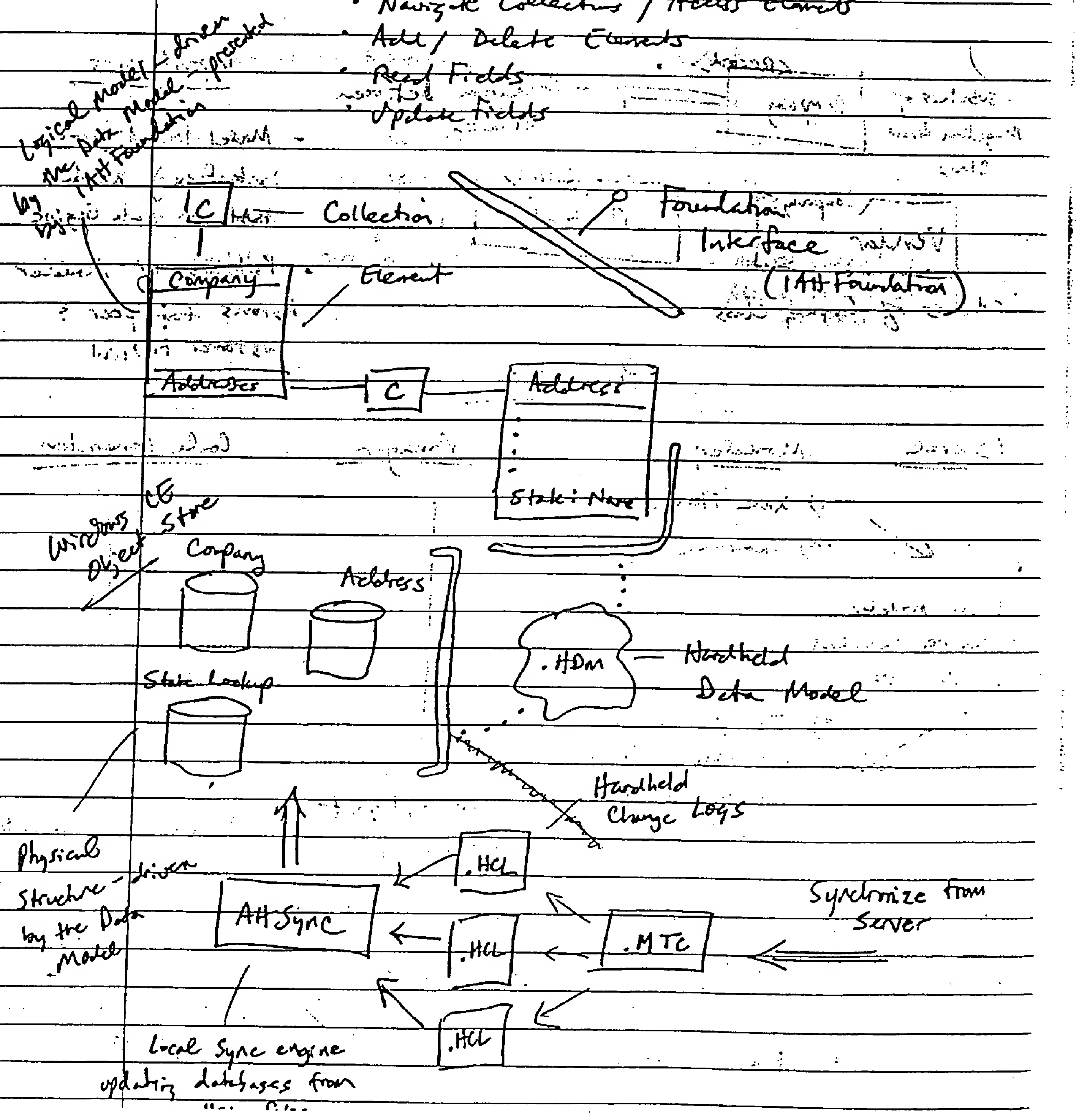
8) Full graphical editor (using properties dialogs from table interface)

23 - Feb - 1999 (Tuesday)

2-23-99

User A

- Navigate Collections / Access Elements
- Add / Delete Elements
- Read Fields
- Update Fields


Logical Model - driven by the Data Model - presented by IAH Foundation
C
Collection
Company
Element
Addresses
C
Address
State : Name
Foundation Interface (IAH Foundation)
Windows CE Object Store
Company
Address
State Lookup
.HDM
Handheld Data Model
Handheld Change Logs
Physical Structure - driven by the Data Model
AH Sync
.HCL
.HCL
.HCL
.MTC
Synchronize from Server
Local Sync engine updating databases from

2-23-99

23-Feb-1999 (Tuesday)

Data Store Server

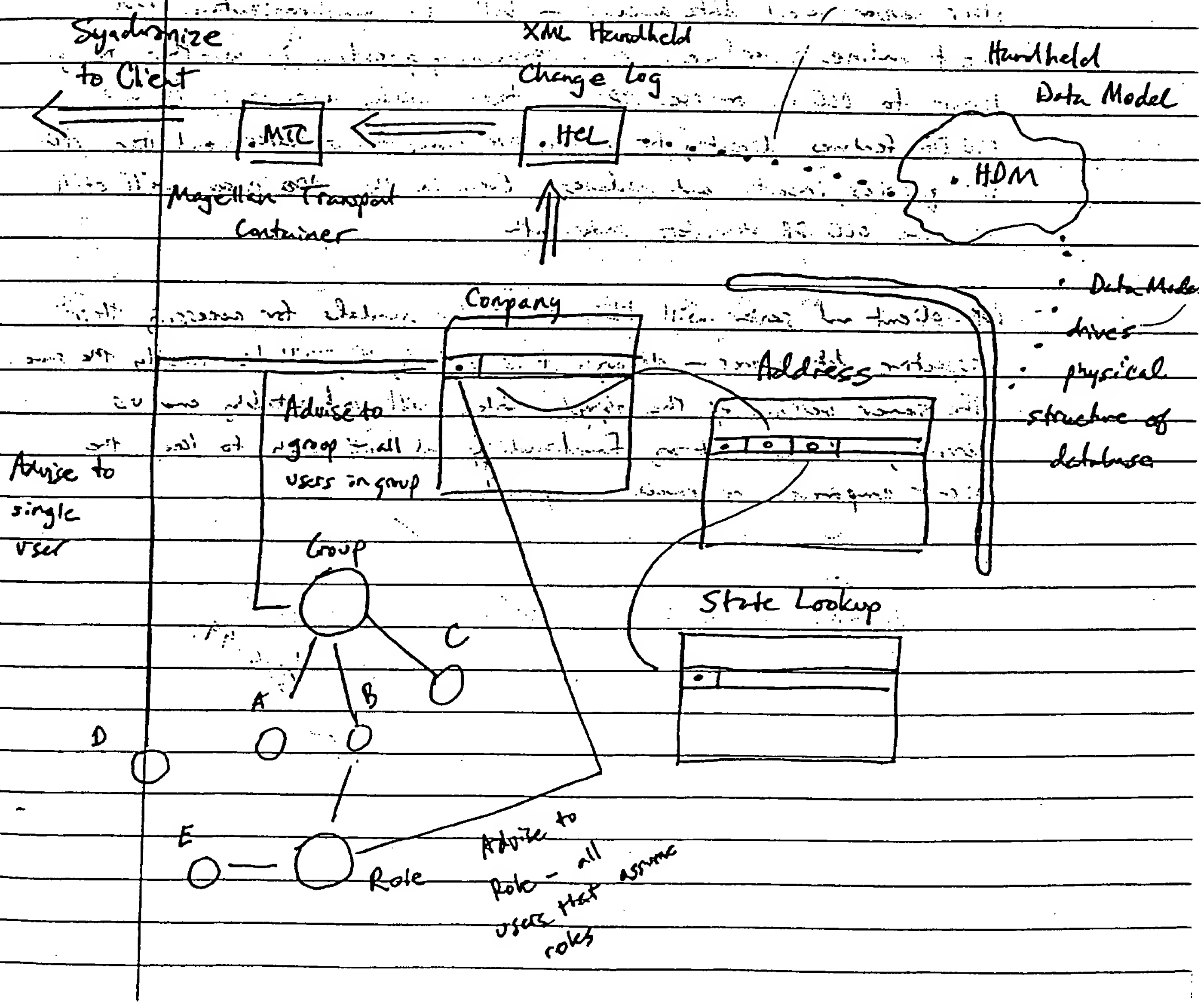
Outgoing

- Distribute New User
- Redistribute Reset User
- Distribute Additions
- Distribute Field changes
- Distribute Deletions

Incoming

- Apply updates to fields
- Insert new elements
- Delete elements

Version of Data Model stored in HCL



- Data Model drives physical structure of database

05-March-1999

- discussed with John Barrett today revisions to IAHDS Catalog design in order to support not only using data model to create DB structure BUT also reading an existing DB structure and generating an HDM data model from it. This will be important as we seek to support handheld applications that connect and sync to existing databases.

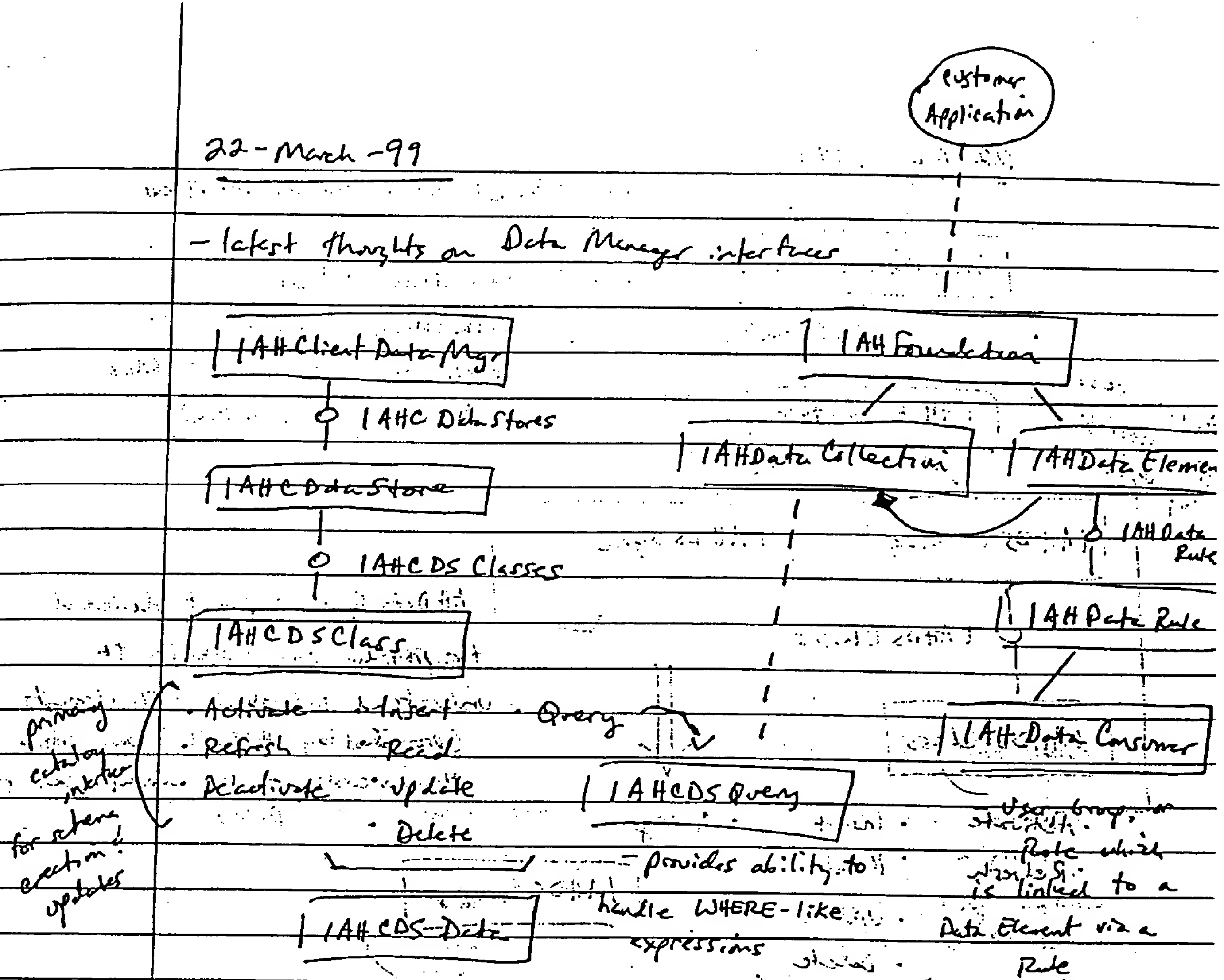
- Committed to using OLE DB for both Catalog component and other server-based data access — both for synchronization and direct online interface. Only remaining question is whether to go direct to OLE DB or use ADO on top. Likely answer is to use OLE DB features directly for schema creation and updates and use ADO for queries, inserts, and updates. Even in the ADO case, we'll still use the OLE DB Providers underneath

- Both client and server will likely use similar models for accessing their respective data stores — although it is unlikely they will be exactly the same. The server version of the object model will undoubtably end up servicing a larger set of functionality and well wish to keep the client components as small as possible

3-5-99

22-March-99

- latest thoughts on Data Manager interfaces

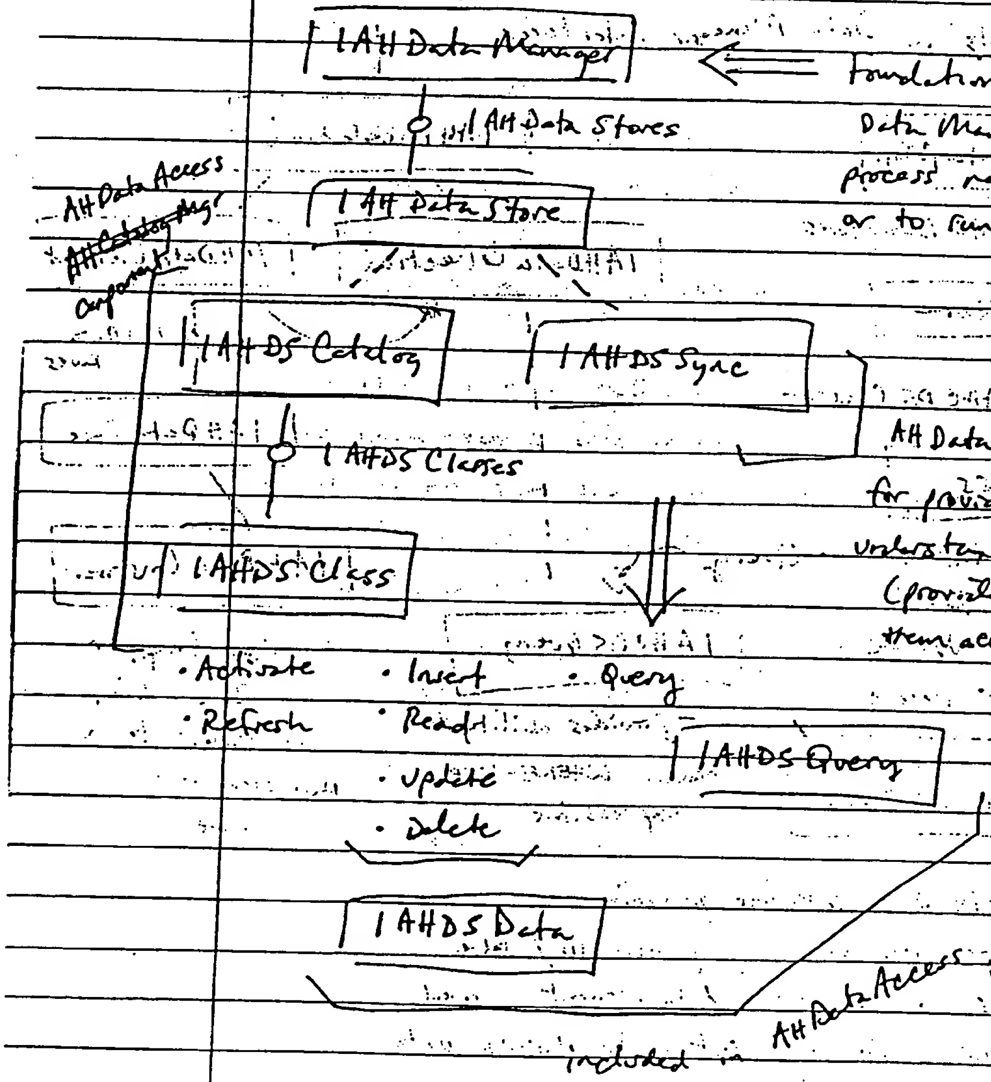


- IAHCDS Data interface primarily encapsulates an array of fieldID / value pairs for a particular data element derived from an ADO recordset. An IAHCDS Data interface pointer is a primary input parameter to the ::Insert, & ::Update methods of IAHCDSClass and an output parameter (actual) to ::Read

3-22-99

22-Mar-1999

- Server-based Data Manager interfaces



Foundation will signal Data Manager in order to process new incoming HTL files or to run batch of outgoing

AH DataSync component designed for providing core capability to understand individual HTL elements (provided by HTL OL) and process them according to the current set of Rules

3-22-99

3-23-99

23-Mar-1999

- Meeting with Ray to review first prototype of Data Modeler
- Suggested minor changes to graph construction, including separation of keys & fields, fonts, colors
- Ray plans to start on left-pane of interface which shows the lists of classes and connections

- Meeting with Omid to review server installation and update process:

Secondary Server

Brand new server

(A)

1) AtColonist stand-alone EXE executed on NT machine

2) Colonist contacts Primary (using server and credentials gathered from user)

3) Primary returns server type & catalog of components

4) Colonist initializes environment

5) Colonist downloads core components in library

(B)

1) Colonist performs full install (file copy, register, MTS stuff, etc...)

2) Colonist starts foundation

3) Colonist shutdown

(C)

Update Existing Server

1) Foundation receives deployment signal

2) Starts existing Colonist (Foundation STILL running)

3) Colonist Executes (A)

4) Colonist signals foundation that (A) is complete.

(D)

1) Foundation decides it is time to update

2) Unpacks & deploys new colonist (as needed)

3) Starts Colonist.

4) Foundation shuts down

5) Colonist executes (B)

23-Mar-1999 3-23-99

Primary Server

New Primary Server

1) Pop in CD & run Installer
2) Installer gets path & credentials from user
3) Installer puts files on disk & writes some form of Colonist signature to disk or registry
4) Installer unpacks Colonist & executes program
5) Colonist Executes Ⓐ & Ⓑ

Update Primary Server

1) Pop in CD (or contact website) and run Installer
2) Installer detects existing
3) Installer writes files to disk
4) Installer unpacks Colonist and runs program
5) Colonist Executes Ⓒ (starting at step 2)

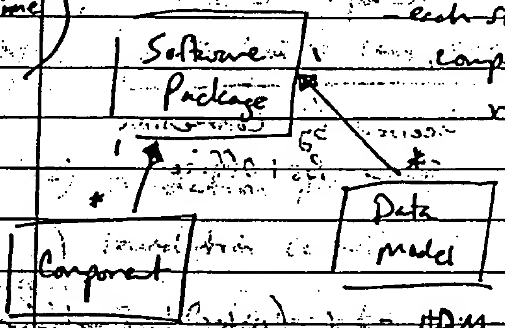
Outstanding Issues

- ~~Need additional semantics / support in Library to support core component installation (deployment)~~
- Need to flush out process of deployment initiation (through Admin) and messaging between servers
- Need to design Rollback process
  - Primary initiated (using Admin)
  - Local server initiated (Control Panel applet)
- Colonist Ⓐ process has to account for both Primary and Secondary cases
- Colonist will require GUI in new install case (for providing credentials and reviewing progress)
- Does the Colonist need to be an NT Service in the case where updates need to occur while the server is logged out?

05-April-1999

- working on design for Application management

Design-time View

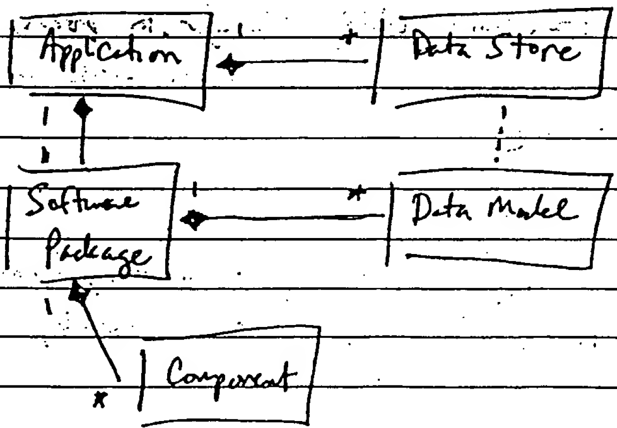


- each Software Package contains one or more components and Data Models. Subsequent versions of the Package will likely have different constructions.

- EXE or DLL required for installation of Software Package
- Added and Updated by the Component Manager component of Magellan Developer

HDM required by components of the Software Package

Production View



Applications are created at runtime, based on a particular Software Package version. The number of data models required by the Software Package dictates linkages to Production data stores

4-5-99

4-13-99

13 - Apr - 1999

1 - meeting with Omid about MTC transfers (for various purposes) from server to server

Ⓐ Incoming Data from Client

1) Type = Incoming Data
Destination = Data Store
MTC
→ Received by Connection Server Post Office

Each MTC will have embedded type and destination ID

2) Foundation picks up Received (notes type and destination)

3) Foundation sends MTC to Data Store Server (on behalf of Data Store ID)

4) Foundation on Data Store Server receives MTC (notes type and destination)
- Unpack MTC → MFD (containing HTC)
- Posts to repository for Data Store (by ID)
- wake up Data Manager

5) Data Manager instantiates IAHSyncManager to process incoming HTC

Ⓑ Deploying Data Model to Data Store

1) Type = Data Model Deployment
Destination = Data Store
MTC
→
- Sent by Primary Server
- Received by Data Store Post Office

2) Foundation on DS Server picks up Received (and notes the MTC type and destination)

3) Foundation puts contained MFD in repository (with version)

4) Signals Data Manager

13-Apr-1999

(C) Component Update for Server

1) Type = Component Update
Destination = (any) server
MTC

→ Received by any Secondary server Post Office using special GUID

2) foundation picks up Received

3) foundation commits MFDs to repository

[• packing slip
• components]

GUID for each component

4) Signal colonist

(D) Application Update for Client

1) Type = Application Update
Destination = Server (Connection)
MTC

→ Received by Connection Server Post Office

2) Foundation picks up MTC

3) Foundation posts MTC to Repository

[• App Deployment Manifest containing each user to be deployed <u>and</u> a list of their existing components]

4) For each user, the Foundation
- builds MTC
- posts to Post Office for user

Type = Application Update
Destination = User (client)
MTC

4-13-99

16-Apr-1999 Friday

- Conference Call w/ FedEX

460 K per data slice
600 slices

- windows CE / Visual C++
- application in forth now – EPSA Table & application
- units connected to docking station connected to PC
- Very interested in our abilities with regard to tracking deployments (who got what, when, etc...)
- Hardware design ongoing (working w/ Eclipse ?)
- Have software requirements for new application

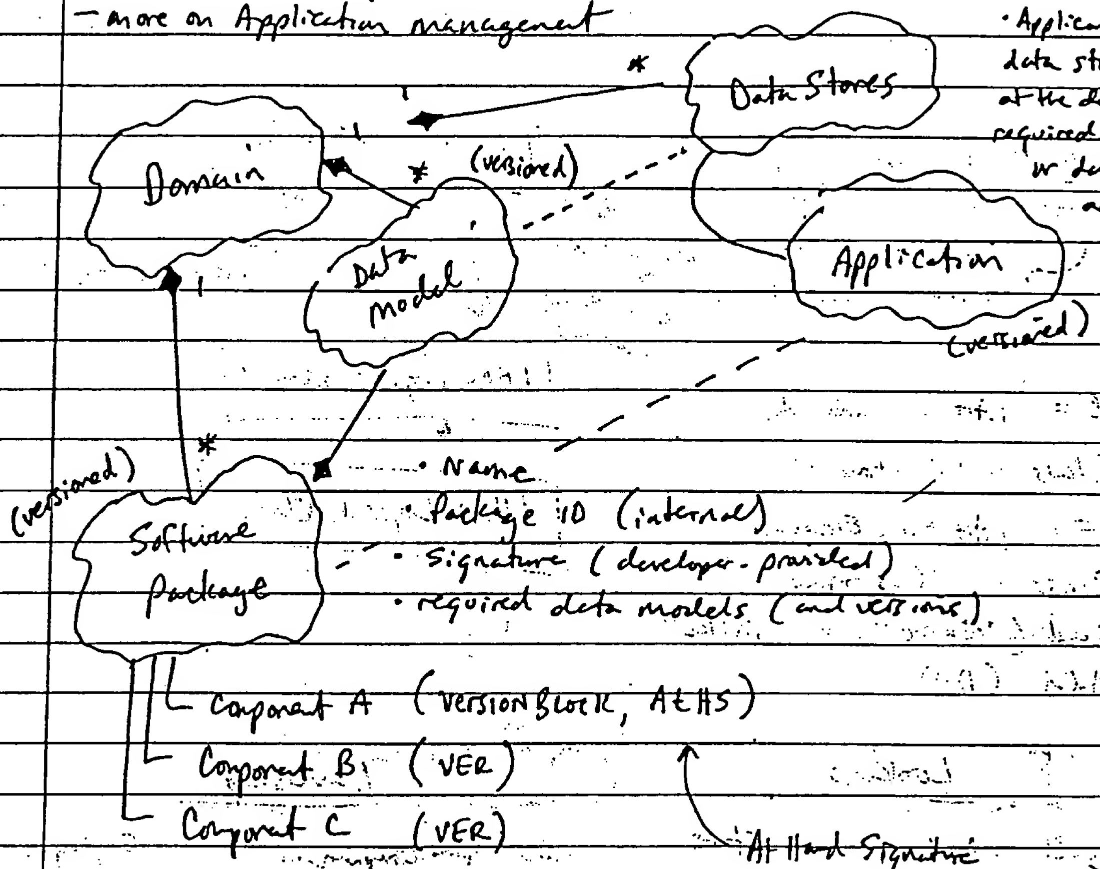
- Conference call w/ Concur

- Discussed plans to rollout XMS to handhelds
- Concur wants a solution which can give them market presence with a minimum of resources spent
- Entire system is Windows NT (ASP, COM, etc..) based
- They liked idea of us taking responsibility for building all or part of their handheld client
- Discussed AvantGo as competitor
- Concur folks will discuss options and get back to us

4-16-99

19-Apr-1999 Monday

- more on Application management



- Name
- Package ID (internal)
- Signature (developer-provided)
- required data models (and versions)

Component A (VERSIONBLOCK, AtHS)
Component B (VER)
Component C (VER)

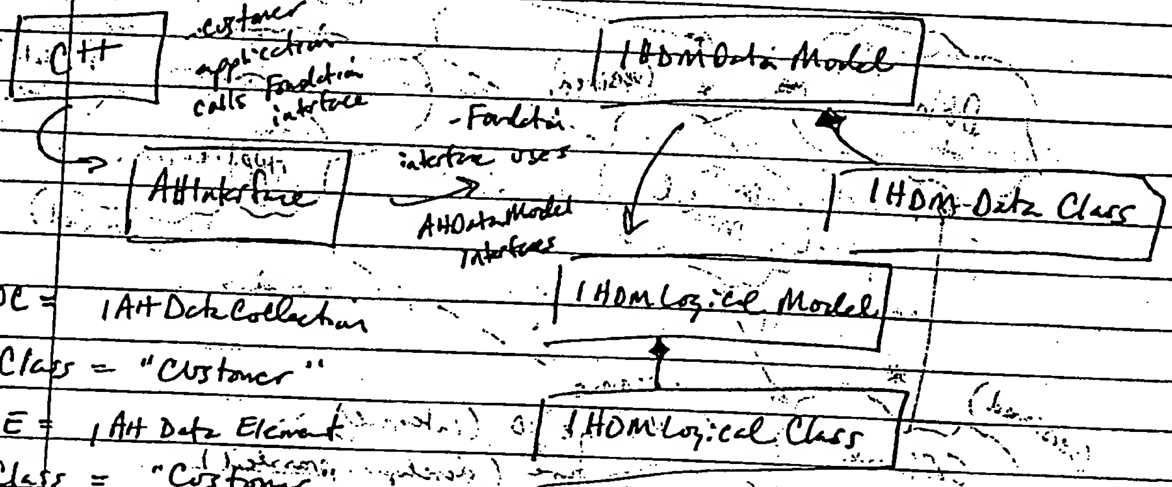
- Application linked to data stores which are of the data model required by the app or data stores which are prepared to be updated

4-19-99

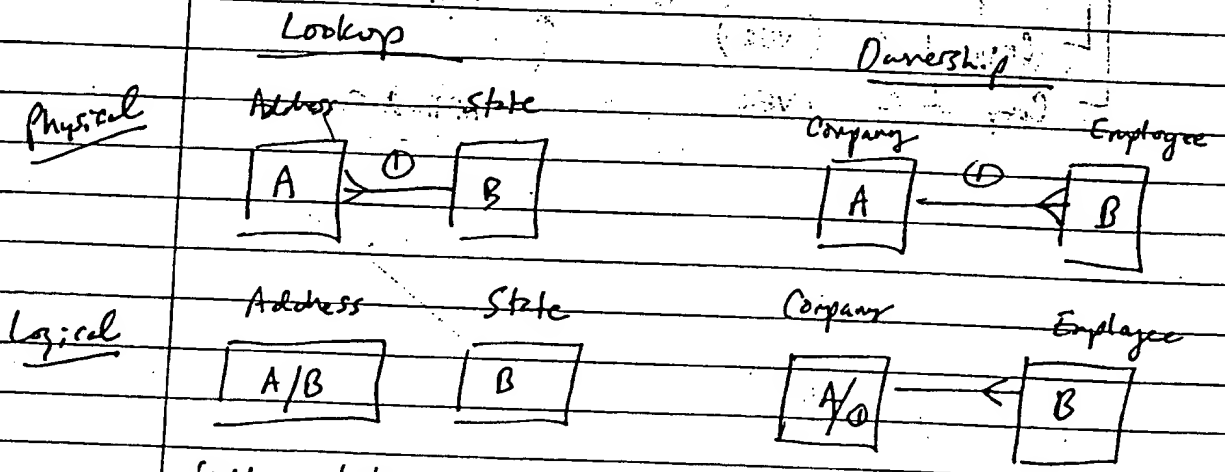
4-22-99

22-Apr-1999

- Discussing conversions between physical & logical Handheld-Data Models


C++
customer application calls Foundation interface
AHInterface
-Foundation interface uses AHDataModel Interfaces
IHDMData Model
IHDM-Data Class
IHDMLogical Model
IHDMLogical Class

```
Set DC = IAHDataCollection
DC.Class = "Customer"
Set DE = IAH Data Element
DE.Class = "Customer"
DE.Field ("Name") = "Mike"
DC.Add (DE)
```



-fields of both 'Address' and 'State' record are present in logical 'Address' class

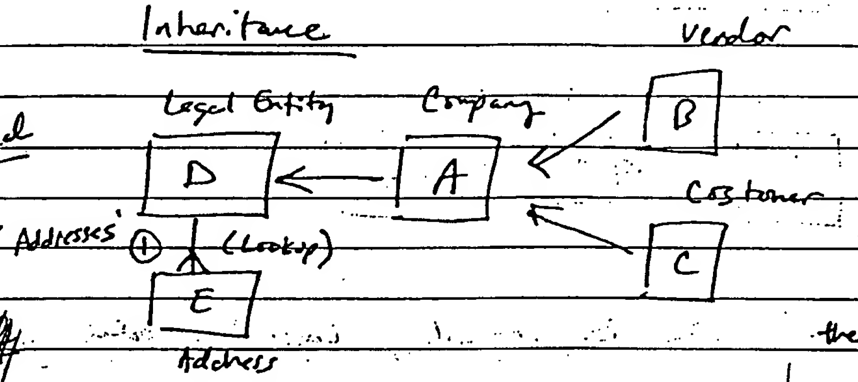
- Special field (with name of connection) is available in logical 'Company' class which returns IAHDataCollection for 'Employee' class

4-22-99

22-Apr-1999

Inheritance

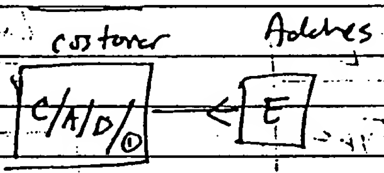
Physical



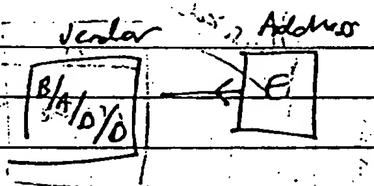
in physical model, 'Legal Ent. 'Company', 'Vendor', and 'Costomer' classes are all linked via the 'Legal Entity' primary key

Logical

Customer



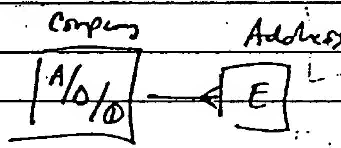
Vendor



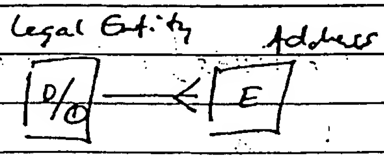
~~avail~~ fields of 'Costomer', 'Company', and 'Legal Entity' are available in logical 'Costomer' class. Also available is collection field (Addresses)

'Vendor' logical class presents same base fields as 'Costomer', ~~except for~~ but not the specific physical 'customer' fields ('B' fields instead of 'C')

Company

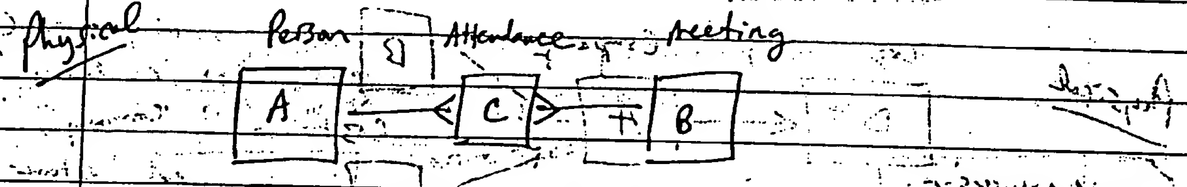


Legal Entity

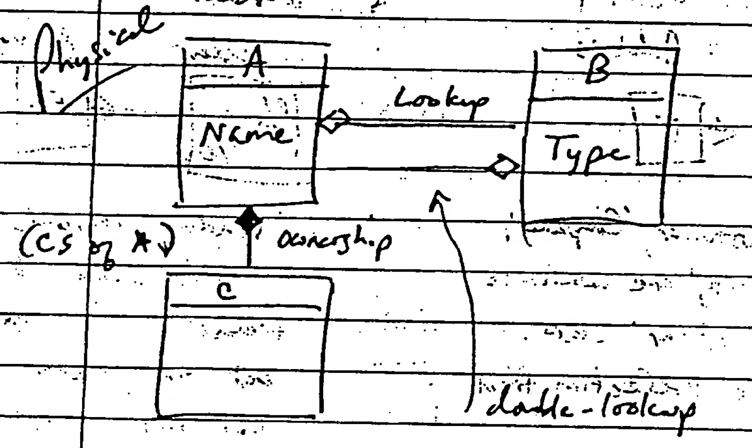


18 - May - 1999
~~22 April 1999~~

Associative

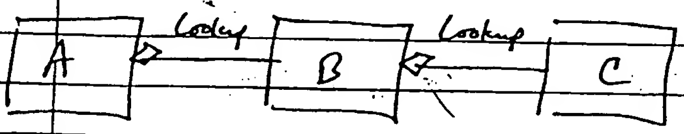


- Discussion with Ray regarding, Logical Model generation
- introduced concept of . Element to handle double-lookup case



Logical

A	
Name B. Type	Fields
B	Elements
C's of A	Collections



SELECT [A Fields] [B fields] [C Fields]

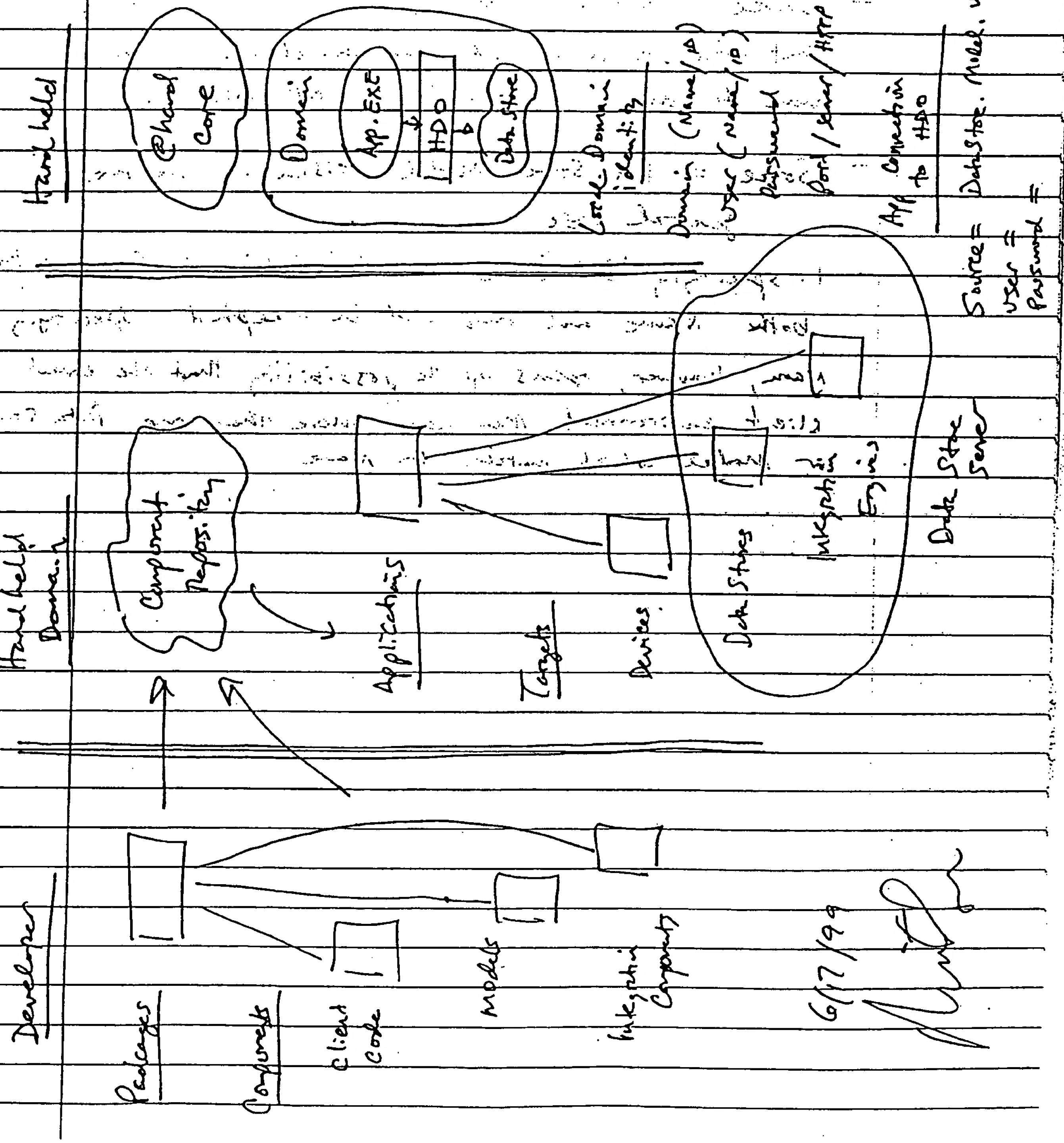
FROM A, B, C

WHERE [A-to-B Join] [B-to-C Join]

5-18-99

17 - June - 1999
~~26 - April - 1999~~

- Discussion with Ray & Orion regarding attributes required for client application to connect to HDO



6/17/99

17-June-1999 (Continued from previous page)

- Client application needs to provide 'Source' to connect to HDO

- Source = Model.Version will work, assuming that only one local data store matches. If there is more than one, then the 'Open' call will fail.

- Source = Data Store.Model.Version is the most general case

In specifying 'Data Store' and 'Model' in Source setting, both Name and GUID with be accepted. Specifying Name, however, opens up the possibility that the current client environment may have more than one Data Store or Model which matches the Name

Introduction

The purpose of this document is to capture high level requirements for the new @hand software product code-named Magellan. The focus is on the capabilities needed by customers and users. It is expected that each approved product feature will be further detailed in one or more Software Requirements Specification (SRS) documents.

The Magellan product is designed to provide a highly effective platform for developing, deploying, and maintaining applications for mobile handheld devices. Magellan will be based on a Windows NT-based Application Server and will support mobile devices running either the Windows CE or Palm OS operating system. The product will center on a set of highly interconnected modules that make up an enterprise-enabled framework. The Magellan framework will be cohesive—but also highly extensible—supporting the integration of one or more lightweight handheld applications with multiple back-end systems. Deployed effectively, the Magellan platform will enable a wide-range of task-specific applications—many of which will be built for classes of users that have not been previously accessible to application developers.

Contained in this document are the specific product-level requirements for each of the modules that will make up the Magellan product. In addition to these requirements, the Magellan project will focus on creating a product that:

- Eliminates the vast majority of unnecessary effort that is currently spent creating or integrating the foundation technologies necessary for successful development, deployment, and management of handheld applications;
- Further streamlines handheld application development through implementation of an object-based data model and support for common development languages;
- Simplifies and speeds integration with multiple concurrent back-end systems;
- Provides applicability to the widest number of market segments—as well as technology that can operate in the widest practical variety of database, handheld, and enterprise environments; and
- Delivers utmost stability to minimize failure generated data problems—as well as performance that can scale to support large numbers of enterprise users.

Background

Market

Within the next decade, handheld computing devices are expected to surpass personal computers in widespread acceptance and use. In terms of units shipped, IDC conservatively predicts that the market will grow from 3.1 million in 1997 to over 13.7 million in the year 2001. For 1998, Gartner Group predicts a 148% growth in unit sales. In terms of absolute dollars, analysts Frost and Sullivan predict that the handheld market will grow to $1.77 billion by the year 2002.

Although the growth numbers are impressive, the real market—and market potential—for handheld devices has only begun to emerge. To date, the majority of handheld devices have been purchased by individuals rather than by IT departments, primarily for personal productivity. Once in use, these individuals have then called on IT to integrate their devices with email, calendar, contact management and other desktop applications. In a sense, handheld devices have thus mounted a steady, yet backward invasion of IT, one achieved through personal motivation—as opposed to strategic initiative.

A much larger market involves the deployment of handheld devices to increase enterprise productivity. Moving beyond sporadic gains for single users, enterprise productivity focuses on the utilization of handheld devices to meet strategic objectives—ranging from the departmental to the corporate level. An enormous potential exists to integrate handheld devices with vital back-end systems ranging from field service to sales automation, enterprise resource planning, scheduling, inventory control, customer management and many others. For business managers, handheld devices present a low-cost platform to leverage the benefits of corporate information resources more broadly throughout the organization.

At the same time, the simplicity and flexibility of handheld devices as a platform and form factor continue to drive new and innovative deployments to entirely new classes of users within the organization. In illustration of this point, the New York Stock Exchange (NYSE) recently announced an initiative to replace paper-based floor trading with handheld devices that relay transactions to systems running the Big Board. According to the NYSE, handhelds provide greater accuracy between traders on the floor and, successively, greater accuracy on the Big Board (and to the market in general) through nearly real-time updates. This type of application would not be possible with desktop or laptop computers because of their inherent complexity and size—nor would it have been possible without an enterprise-level commitment to the initiative.

The New York Stock Exchange also constitutes an example of specific-purpose computing, the deployment of a task-specific applications on simple, low-cost handheld devices. In contrast to the general-purpose computing model that most desktop computers represent, specific computing focuses a few target functions into lightweight applications for a particular class of end-users. Since these functions typically derive from one or more back-end systems, this new class of applications provides business managers a means to increase their investment return by extending core systems further into the organization. For end users, these lightweight, job-specific applications enable them to work more efficiently—without the complexity and learning-curve normally associated with existing back-end systems.

Importantly, great handheld applications work equally well whether online or offline, enabling users to employ the benefits of corporate IT at the places where business transactions occur. This is in stark contrast to Web-based systems that are crippled once the tether is removed. In terms of business impact, applications that interact with users—when mobile or online—offer tremendous benefits for both enterprise productivity and customer service.

One example is field service. In addition to clipboards and pagers, consider the deployment of a task-specific application on a handheld device to an entire workforce of field service personnel. Along with customer contact info, issues and route map, this application would provide the ability to:

- review service history;
- examine component diagrams;
- utilize flow charts for troubleshooting and routine maintenance;
- order parts;
- quote cost & delivery dates;
- schedule a return visit; and
- forward sales leads to the appropriate point within the organization.

Independently, these tasks typically draw from an array of back-end systems, including customer service, accounting, scheduling, order entry, sales automation, and supply chain applications—potentially even with third-party suppliers and information systems via company-to-company extranets. This complexity, however, remains transparent to the end user as it is encapsulated within an easy-to-use application that was *specifically built* to address that end user's role within the organization. In terms of functionality, the user's application is not simply an electronic version of the same documents they use to carry around on paper. Instead, it is interactive, intuitive, and even heuristic—an application built to make the end-user more productive, not just better connected.

The traditional barrier to these benefits has always been the high cost and complexity associated with developing, deploying and managing handheld applications that are truly integrated with multiple back-end systems. To date, just about every enterprise deployment of this nature constitutes a custom in-house effort involving hundreds of thousands *or millions* of dollars and man-years of development time. Moreover, even for organizations that have been willing to undergo this expense and risk—FedEx, UPS and NYSE among others—the lack of a development framework has meant that the first-generation handheld applications actually deployed have been entirely static and incapable of evolution. This last fact sadly deprives business managers of the ability to alter strategic applications as competitive needs dictate.

To address these issues of cost, complexity and agility, Magellan provides a robust and intelligent platform enabling IT organizations to rapidly build, deploy and evolve strategic handheld applications that are integrated with existing back-end systems. This platform will form the cornerstone around which companies build true enterprise productivity now and into the future.

Target Users

Target customers of the Magellan product will be organizations that seek to extend existing standalone applications to handheld users. This list will include:

- . Companies (and departments) that are building internal proprietary applications;
- Independent Software Vendors (ISVs) who are writing applications primarily for resale;
- Solution-oriented Systems Integrators that develop custom applications for clients; and
- Original Equipment Manufacturers (OEMs) that build applications that add value to their hardware.

All of these organizations have one thing in common—that is that they have already identified and implemented one or more strategic applications and intend to use the simplicity, low cost and small form factor of handheld devices to extend these application to new classes of users within the organization and/or the places where business transactions occur.

Potential ISV customers include:

- Sales Force Automation (SFA) companies like Siebel Systems, Vantive, SalesLogix, Pivotal, Onyx, Aurum (Baan) and Moss Micro;
- Enterprise Resource Planning (ERP) companies like SAP AG, J.D. Edwards, PeopleSoft and Baan;
- Field Service companies like Astea and Clientele; and
- Technical Support companies like Clarify and Remedy.

A few vertical examples include:

- Utility companies like Southwestern Bell, Consolidated Edison, and Time-Warner extending field service applications to in-house and third-party field service technicians;
- Transportation companies like FedEx, UPS, and CSX extending transportation management systems to drivers, warehouse and yard managers;
- Health and medical organizations like Hospitals, Physician groups, and HMOs improving patient care and efficiency by extending systems to doctors, nurses, technicians, and EMS personnel.

Key User Needs

The business and technical requirements of this platform are straightforward. For business managers, the initial requirement is speed. A handheld application platform should facilitate project development cycles that are measured in man-weeks rather than man-months and man-years. Shorter development cycles should then yield applications that are deployed to the users as close as possible to the time of maximum impact. This, in turn, enables businesses to utilize IT assets to respond nimbly to competitive, customer and internal needs, in addition to reducing overall development cost and risk.

The second major requirement calls for a high degree of user-acceptance. Experienced business managers know that end users prefer applications that are highly targeted to their needs and simple to use. Consequently, successful application developers are those that can focus primarily on the content, presentation, and impact of the application—not the underlying low-level services that end-users never see. At the same time, successful application developers know the value of continual feedback. They can more easily ensure that an application is successful when they react quickly to ever-changing user needs. The largest single factor governing their ability to react is not necessarily their ability to change the code—but rather their ability to seamlessly deploy that code out to a distributed set of clients. Therefore, a handheld platform has to provide a strong foundation that supports both a rich set of services and the technology to manage change.

Business managers also require strong integration with the widest possible array of back-end systems on a variety of platforms. These systems—ranging from front office applications like product configurators to back office applications like enterprise resource planning—comprise strategic assets within an organization. A handheld platform should provide the ability to intelligently exchange data and information between these systems and handheld applications with a minimum of development effort and time.

A fourth major requirement of business managers dictates a high degree of application flexibility. Although considered strategic assets, typical first-generation handheld applications remain static, effectively incapable of evolution once deployed. In terms of return on investment, these applications yield their best results immediately after release, but quickly diminish as internal needs evolve in response to changing customer needs and competitive pressures. Against this backdrop, business managers seek an environment that permits the iterative evolution of handheld applications as needed.

Technology managers also hold a short but important list of requirements. Chief among them is simplicity. Currently, roughly ten percent of the development effort for handheld application actually focuses on the application itself—*i.e., what the application does*. Instead, nearly 90% of the effort and cost revolve around implementing complex system-level activities such as the communications, synchronization and data access that support the application. Technical managers will embrace a platform that transparently manages these latter elements, enabling developers to focus on the application itself instead of the underlying plumbing.

Another requirement is a versatile development environment, permitting work in the language appropriate to the needs of a particular project—C++, Visual Basic, Java, etc. Or, by corollary, the language most appropriate for the IT organization based on in-house skills and talents. Application servers that limit choices in this regard always fight unwinnable uphill battles against both legitimate and "religious" concerns.

Manageability remains the third pivotal requirement for IT managers. Specifically, the platform must integrate an array of operating systems and devices with existing network and directory services within IT. At the same time, the platform must simplify the remote deployment and installation of handheld applications, data and schema—both for initial rollouts and subsequent iterations.

Finally, the nature of IT's mission—supporting large numbers of users on mission critical applications—requires that scalability and reliability be the hallmarks of any enterprise solution.

Assumptions

The Magellan product will be delivered as a set of modules that can be mixed and matched depending on the customer' s needs. The first version of the whole product will be totally self-contained, but integrated with additional @hand products over time.

@hand assumes that Windows NT will continue to grow and remain the dominant platform for the deployment of mission-critical application servers for at least the next three years. Another significant assumption is that Windows CE and Palm OS will continue to emerge as increasingly attractive platforms for the deployment of targeted applications for an increasingly mobile workforce.

@hand also assumes that in addition to simply extending existing third-party applications to handheld devices, IT departments will also increasingly seek to evolve the handheld version of the applications to offering more and more task-specific subsets of functionality from **multiple** back-end systems. In other words, handheld applications will be increasingly oriented around the user's role within the organization as opposed to being solely an extension of a single back-end system.

To meet the expected delivery dates, it is critical to have adequate resources including knowledgeable personnel, equipment, and space. The technical skills required to undertake a project of this type are in high demand in today' s market rendering it a challenge to hire or contract with appropriately skilled personnel. The company assumes that it will be successful in recruiting and retaining such personnel in order to meet its stated deadlines. Due to the significant and rapid changes in technology, @hand will continue to keep abreast of new advances and modify the requirements as necessary.

Links

Other documents which should be reviewed in conjunction with this one:

- The Magellan Architecture is captured in a separate document entitled, **Project Magellan – Complete Architecture**. Please reference that document for issues in this area.
- The complete Magellan Project Plan and Implementation Roadmap are captured in separate document entitled, **Project Magellan – Project Plan**. Please reference that document for issues in this area.
- The model for database synchronization utilized by Magellan is captured in a separate document entitled, **Project Magellan – Synchronization Model**. Please reference that document for more detailed specifications and design.
- The model for data access from client applications utilized by Magellan is captured in a separate document entitled, **Project Magellan – Data Access Model**. Please reference that document for more detailed specifications and design.
- Requirements for individual modules are captured in various Software Requirement Specification (SRS) documents. Please see those documents for more details on specifications and design.

1. Application Server Features

The cornerstone of the Magellan product is its Handheld Application Server, providing the central authority for all application, database, and systems management functions. The features listed below describe the particular capabilities of the Application Server that deliver benefits to users. These capabilities provide the fundamental basis for all product definition, requirements, and project management.

1.1. General

The following requirements detail the general constraints that govern the development of the Magellan Application Server, including operating systems support, high-level module descriptions and platform-level dependencies on other software products:

PR-1.1.1 The Magellan Application Server will initially support Windows NT Server, Version 4.0, installed on x86 and Alpha-based servers, with Service Pack 4 (SP4) applied.

PR-1.1.2 Magellan will support Windows 2000 (NT 5.0) Server and Advanced Server within 90 days of its general release by Microsoft.

PR-1.1.3 Magellan will require Microsoft Transaction Server 2.0 as its target environment for hosting internal core COM components and customer-authored Integration components.

PR-1.1.4 Magellan will require the installation of Microsoft's Internet Information Server (IIS), Version 4.0 to support browser-accessible administration functions.

PR-1.1.5 **Magellan Foundation Manager**: The Magellan Application Server will support a dedicated NT process referred to as the Magellan Foundation Manager that will act as the base-level executive for the entire platform. Magellan will require installation of its Foundation Manager on any server that will support Magellan Application Server processes. All other process-level modules will support the ability to be remotely deployed by the Foundation Manager to other distributed Magellan servers. Consequently, it is expected that each instance of the Foundation Manager will communicate with various other instances—running on other NT servers—in order to support basic load balancing, functionality distribution, and failover options.

PR-1.1.6 **Magellan Messenger**: Magellan will support a dedicated NT process referred to as the Magellan Messenger that will handle all connectivity between the Application Server and supported handheld devices. In addition, instances of the Magellan Messenger will also manage connectivity between individual Magellan processes running on various servers and between Magellan Developers and Magellan Application Managers.

PR-1.1.7 **Magellan Directory**: Magellan will support a dedicated NT process referred to as the Magellan Directory that will provide necessary authentication and directory services to other Magellan modules, such as the Messenger and Data Engine. It is expected that the Magellan Directory will support and perhaps utilize the Active Directory interfaces when available in NT 5.0.

PR-1.1.8 **Magellan Data Engine**: Magellan will support a dedicated NT process referred to as the Magellan Data Engine that will handle all connections to one or more Magellan databases (both runtime and development), support synchronization functions, and manage all versions of the

Data Model for one or more Magellan databases. Magellan Data Engines are expected to support data for both individual Magellan Clients and other Magellan Data Engines which relate hierarchically to it.

PR-1.1.9 **Magellan Administrator**: Magellan will support a two-tiered mechanism for administration and configuration of the Magellan Application Server. The interface will include a Microsoft Management Console (MMC) interface for configuration of low-level services, and a browser-based interface for higher-level functions.

PR-1.1.10 **Magellan Application Manager**: Magellan will support a dedicated NT process referred to as the Magellan Application Manager that will support the development, management, and deployment of all Magellan applications. Consequently, the Magellan Application Manager provides the primary interface to the Magellan Component Repository database—via a Magellan Data Engine. Variants of the server-based Magellan Application Manager will also be present in Magellan Client and Developer modules

PR-1.1.11 **Magellan Foundation Interface**: Magellan will support a set of COM-based interfaces for programmatic access directly to the Magellan Application Server and its data. These services will be designed primarily to support the Magellan Integration Server, but will be accessible to other processes as well. Subsequent versions of the Magellan Foundation Interface are expected to support CORBA links.

1.2. Magellan Foundation Manager

The following requirements detail the features of the Magellan Foundation Manager, outlining its support for executive-level management of the Magellan Application Server platform:

PR-1.2.1 The Magellan Application Server will support a dedicated NT process referred to as the Magellan Foundation Manager that will act as the base-level executive for the entire platform.

PR-1.2.2 Magellan will require installation of the Foundation Manager on any server that will support Magellan Application Server processes. The Foundation Manager will be installed as an NT Service to allow unattended operation even while the server is logged-out.

PR-1.2.3 The Magellan Foundation Manager will host and support all other Magellan Application Server processes, including Messengers, Directories, Data Engines, Application Managers, and Integration Engines.

PR-1.2.4 The Magellan Foundation Manager will support the COM-based Foundation Interface for programmatic access directly to the Magellan Application Server and its data. This interface will be designed primarily to support the Magellan Integration Server, but will be accessible to other processes as well.

PR-1.2.5 The Magellan Foundation Manager will support the instantiation, termination, scheduling, and configuration of all hosted processes. All process-level Magellan modules will support interfaces to facilitate this support.

PR-1.2.6 The Magellan Foundation Manager will support the ability to communicate with other Foundation Manager instances running on other NT servers via Magellan Messenger instances.

PR-1.2.7 The Magellan Foundation Manager will support the ability to remotely deploy new and updated versions of process-level modules (e.g., Data Engines) to other distributed Magellan servers that are running a Foundation Manager.

PR-1.2.8 The Magellan Foundation Manager will act as the initial connection point for all handheld-to-server communications, effectively "owning" the IP address and port number for a particular NT Magellan Server. Once initial contact is established, the Foundation Manager will "spin-off" multiple Messenger threads to handle each individual communication session.

1.3. Magellan Messenger

The following requirements detail the Connectivity features of the Messenger module of the Magellan Application Server, outlining its support for connecting multiple Magellan Clients to the central server:

PR-1.3.1 Magellan will support a dedicated NT process referred to as the Magellan Messenger that will handle all connectivity between the Magellan Application Server and supported handheld devices running a Magellan Client.

PR-1.3.2 The Magellan Messenger will manage synchronization-level connectivity between individual Magellan processes (e.g., Foundation Managers, Data Engines, etc…) running on different network-connected servers.

PR-1.3.3 The Magellan Messenger will manage connectivity between workstation-based Magellan Developers and server-based Magellan Application Manager processes.

PR-1.3.4 The only supported protocol for communications between Magellan Clients and Magellan Messenger processes will be TCP/IP. The only information a Magellan Client (or server) should require to initiate communications with a target Magellan server machine is its IP address and configured port number.

PR-1.3.5 A prerequisite for all supported handheld devices is that they first establish a valid TCP/IP connection to the same network as the Magellan Application Server—either through Remote Access Services (RAS) or direct Ethernet connection—before attempting direct connections to the Magellan Application Server. Magellan will only support connectivity on an existing TCP/IP connection.

PR-1.3.6 Magellan will support concurrent connections to multiple Magellan Clients from a single Application Server. The number of simultaneous Client-to-Messenger connections should be limited only by practical considerations, such as server RAM and processor speed.

PR-1.3.7 In order to support connections to multiple clients, Magellan will support multiple Messenger processes or threads running concurrently on one or more servers. In order to scale up to large numbers of handheld clients, Magellan will smoothly support a typical distributed configuration where the Magellan Messenger processes can be located on a separate server from other Magellan Application Server modules and the Magellan database.

PR-1.3.8 The primary purpose of connection sessions between Magellan Clients and Messengers will be the bi-directional transfer of one or more files, either binary or text. In order to promote highest performance and scalability, connection sessions will be as short as possible, with no significant analysis or computation being performed, only authentications and file transfers.

PR-1.3.9 Magellan will authenticate all connections to the Magellan Server using features of the Magellan Directory, including NT domain authentication, encrypted passwords, etc… Magellan Clients will be required to provide authentication parameters before a successful connection can be established.

PR-1.3.10 Magellan will support the ability for authenticated clients to change their password during the process of a connection session. As a safeguard, for any client that changes their password, the old password can still be used for authentication until the new password has been used at least once (or changed again).

PR-1.3.11 Magellan will utilize some form of industry standard compression algorithm (e.g., zlib) in order to improve the performance of all data transfers.

PR-1.3.12 Magellan will support packet size adjustments for individual clients during file transmissions in order to handle a wide range of wired and wireless transmission speeds and quality.

PR-1.3.13 Magellan will support a mechanism to stop and restart individual file transfers that are in-process (e.g., checkpointing). This will improve performance and reduce user frustration by preventing repeated transfer of partial files in low-quality environments.

PR-1.3.14 Magellan will utilize some form of data integrity checking (e.g., checksum) in order to validate successful whole-file transmissions between clients and servers.

PR-1.3.15 Magellan will utilize some form of strong security measures (e.g., DES, Blowfish, etc…) in order to protect the content of actual data transmissions between clients and servers.

PR-1.3.16 Magellan will support the ability to classify Transaction Files according to their content. (e.g., Application Component, Data, Message, Document, Core Component , etc…)

PR-1.3.17 Magellan will support Coordination Links between Transaction Files of differing Content Types in order to control dependent relationships (e.g., Application Component requires certain version of Data Model to work).

PR-1.3.18 Magellan will support Sequence Links between Transaction Files that require chronological ordering (e.g., Transaction Data).

PR-1.3.19 Magellan will support the ability for Messenger clients to download header information for large files in order for clients to possibly avoid large transfers over slow-speed connections. Coordination and Sequence Links between files should prevent downloads of subsequent files that require earlier files which were skipped.

1.4. Magellan Directory

The following requirements detail the Directory Services features of the Magellan Application Server, outlining its support for user, group, role, and server resource management:

PR-1.4.1 Magellan will support a dedicated NT process referred to as the Magellan Directory that will provide necessary authentication and directory services to other Magellan modules, such as the Messenger and Data Engine.

PR-1.4.2 The Magellan Directory will support and perhaps utilize the Active Directory interfaces when available in NT 5.0.

PR-1.4.3 Magellan will support one or more instances of a Magellan Directory Class called **Domain.** A Magellan Domain is defined as an application environment that supports one or more Magellan NT server machines, which support connections to one or more handheld devices, support the delivery of one or more Applications, and manage a set of data stored in one or more Magellan Databases. A Magellan Domain is the root level object of the Magellan Directory namespace.

PR-1.4.4 Each Magellan Domain will support one or more instances of a Magellan Directory Class called **Server.** A Magellan Server is defined as a Windows NT Server machine that runs an instance of the Magellan Foundation Manager for purposes of hosting one or more Magellan process-level modules (e.g., Data Engine, Application Manager, etc...). Each Magellan Domain will have at least one master Server and potentially many secondary Servers.

PR-1.4.5 Each Magellan Domain will support one or more instances of a Magellan Directory Class called **Database.** A Magellan Database is an instance of one of the relational database types supported by Magellan, including Microsoft SQL Server and Oracle, which is used to store runtime data that is of interest to the Applications in the Domain. Each Magellan Database is managed by a Magellan Data Engine.

PR-1.4.6 Each Magellan Domain will support an instance of a Magellan Directory Class called **Repository.** A Magellan Repository is an instance of one of the relational database types supported by Magellan, including SQL Server and Oracle, which is used to store development components. Each Magellan Repository is managed by a Magellan Application Manager.

PR-1.4.7 Each Magellan Domain will support one or more instances of a Magellan Directory Class called **Application** which is the runtime version of a Magellan Component Class of the same name (see the Magellan Application Manager). Each Magellan Application is composed of a set of Components and a Data Model that is deployed to one or more handheld Users to perform a specific purpose for the User.

PR-1.4.8 Magellan will support one or more instances of a Magellan Directory Class called **User.** A Magellan User is defined as a person or process that takes on a unique identity in a Magellan Domain environment, belonging to one or more Groups, and assuming one or more Roles in those Groups. Magellan Users include both persons running applications on handheld clients and agents applications running in the Magellan environment. All Magellan Users have a name which is unique across the namespace of a particular Magellan Domain. All access to the Magellan Application Server by any User is authenticated by password.

PR-1.4.9 Magellan will support one or more instances of a Magellan Directory Class called **Group.** A Magellan Group is defined as a collection of Users and sub-Groups, all of which share a association based on organization, geography, or other application requirements.

PR-1.4.10 Magellan will support one or more instances of a Magellan Directory Class called **Role.** A Magellan Role is defined as a position or job which can be assumed by one or more Users in a particular Group (e.g., Joe Smith plays the role of Manager in the Western Region Sales group)

1.5. Magellan Security

The following requirements detail the authentication features of the Magellan Application Server, outlining its support to prevent unauthorized access to the Magellan Application Server:

PR-1.5.1 **NT Domain Security:** Magellan will utilize Windows NT Domain security as its primary model. All Users of in a Magellan Domain will also need to be members of an NT domain where the Magellan Application Server itself has privileges. This model will allow Magellan Users to utilize the same username and password for both NT and Magellan authentications. With this model, however, the Magellan Application Server itself will require Administrator-level access to the NT domain in which it operates.

PR-1.5.2 **Alternate Security:** Magellan will also support a secondary security model where Magellan Users maintain a username and password which is separate from their NT domain identification. This model will only require that the Magellan Application Server be granted user-level access to the NT domain in which it operates.

PR-1.5.3 Regardless of the security model chosen, all valid Magellan Users will need to be granted access to a Magellan Domain via the Magellan Administration interface.

PR-1.5.4 All security information will be persisted by the Magellan product using strong encryption techniques (e.g., DES, Blowfish, etc…)

PR-1.5.5 **Access Levels:** Magellan will support the concept of security access levels for Magellan Users, Roles, and Groups. Consequently, Magellan will support the capability to grant or revoke various privileges to each access level. These access levels will be utilized by other Magellan processes for various purposes, including the ability to enforce permissions on reading, creating, updating, and deleting data from the system.

PR-1.5.6 Magellan will support a rich logging and audit mechanism for tracking all security-related interactions between systems modules, including successful and failed authentications.

1.6. Magellan Administrator

The following requirements detail the Administration features of the Magellan Application Server, outlining its support for configuration and maintenance of both low and high-level parameters of the system:

PR-1.6.1 **Low-level Administration:** Magellan will support one or more Microsoft Management Console (MMC) Snap-Ins for use in administration of the low-level services of Magellan Application Server, including communications, low-level security, databases, and performance parameters.

PR-1.6.2 **High-level Administration:** Magellan will support Microsoft Management Console (MMC) and browser-based interfaces for administration of higher-level services, including configuration of Applications, Users, Groups and Roles.

PR-1.6.3 The low-level MMC-based Magellan Admin interface will support the configuration of each Magellan Domain, including the locations of the master and secondary Magellan Servers (Foundation Managers) and the process-level Magellan processes (e.g., Messenger, Data Engine, etc…) which are expected to be hosted on each.

PR-1.6.4 The low-level MMC-based Magellan Admin interface will support the configuration of communication parameters for each of the Magellan Servers, including IP addresses, port numbers, timeout values, etc.

PR-1.6.5 The low-level MMC-based Magellan Admin interface will support the configuration of each Magellan Database, including location, authentication parameters, connection settings, etc.

PR-1.6.6 The low-level MMC-based Magellan Admin interface will support the configuration of all low-level logging and system tracing capabilities supported by the various Magellan processes

PR-1.6.7 In the case where NT Domain Security is being used, the high-level Magellan Admin interfaces will support the ability to select from existing NT Domain users to add to the list of valid Magellan Users.

PR-1.6.8 In the case where Alternate Security is being using, the high-level Magellan Admin interfaces will support the ability to add and remove new users to the system, including the specification of usernames and passwords.

PR-1.6.9 The high-level Magellan Admin interfaces will support the maintenance of Roles and Groups, including the assignment of Roles to Groups, Users to Groups, and Users to Roles.

PR-1.6.10 The high-level Magellan Admin interfaces will support the ability to assign access levels to particular Users, Groups, and Roles.

PR-1.6.11 The high-level Magellan Admin interfaces will support the ability to grant access (or exclude access against a grant) to particular Magellan Applications for single Users, all Users in a particular Role or Roles, or all Users in a particular Group or Groups.

PR-1.6.12 The high-level Magellan Admin interface will support the administration and configuration of each Magellan Repository, including location of the database, maintenance of development privileges for selected Users, etc…

1.7. Magellan Application Manager

The following requirements detail the Application Management features of the Magellan Application Server, outlining its support for the development, management, and deployment of applications to handheld devices:

PR-1.7.1 Magellan will support a dedicated NT process referred to as the Magellan Application Manager that will support the development, management, and deployment of all Magellan Applications within a particular Magellan Domain.

PR-1.7.2 A key function of the Magellan Application Manager will be to support connections to the Magellan Repository, a dedicated database for storing instances of supported Magellan Component Classes.

PR-1.7.3 Magellan will support one or more instances of a Magellan Component Class called **Application.** Each Magellan Application is constructed using the Magellan Developer interface and encapsulates one or more instances of other supported Magellan Component classes.

PR-1.7.4 Magellan will support a Magellan Component Class called **Program** which is an Application Executable for either the Windows CE or Palm OS platform

PR-1.7.5 Magellan will support an Magellan Component Class called **Extension** which is a library component (e.g., DLL) that contains code that is designed for use either by a Program component or by other applications on the target handheld.

PR-1.7.6 Magellan will support an Magellan Component Class called **Library** which is a simple data file that can be used by Program and/or Extension components. An example of a Library component is a configuration file required by the application.

PR-1.7.7 Magellan will support the ability to link various component instances together so that one is not deployed without the other(s).

PR-1.7.8 Magellan will support a Magellan Component Class called **Data Model** containing a set of Tables, Columns, and Connections that describe the structure of a particular database. Magellan will support Applications that have no Data Model, Applications that contain one or more Data Model components, or Applications that are linked to another "master" Application that maintains a Data Model.

PR-1.7.9 Magellan will support an extension to the Magellan Developer product called the Magellan Modeler for maintaining Data Model components.

PR-1.7.10 Magellan will support a full version control capability in the Magellan Developer product that allow developer users to check-out (lock), check-in, and retrieve individual versions of Magellan Components

PR-1.7.11 Magellan will support a Magellan Component Class called **Package**, which "lives" inside a particular Application instance. Package components specify a "pinned" set of the application's components, each of a particular version. Before the first deployment to users, only one version of a particular package is present "inside" an application. But as an application is deployed multiple times to users, multiple versions of a particular Package will be present in the application's configuration. Magellan will enforce that later versions of packages will always contain versions of components which are the same or later than those in previous packages.

PR-1.7.12 **Deployment:** Magellan will support the capability for developers to deploy a Magellan Application (i.e., the latest version of that Application's Package and the versions of the Magellan Components encapsulated by the Package), either immediately or at a scheduled time to a target Magellan Foundation Manager. Upon delivery, the Foundation Manager will prepare the Application for download by Magellan Client users that have been granted access to it. Preparation will include the pre-processing of required application components, data model, and data in a highly compressed package that enables immediate transmission of updates to connected handhelds.

PR-1.7.13 Magellan will support a Magellan Component Class called **Integrator** which is a Visual C++ or Visual Basic DLL for deployment to the Magellan Integration Server.

PR-1.7.14 Magellan will support the ability to distribute new version of Magellan Integrator components from a particular Magellan Foundation Manager to connected Magellan Integration Servers. These Integration Components will be included in an Application's Package in a manner similar to how Program or Extension-class components are handled.

1.8. Magellan Data Engine

The following requirements detail the Data Management features of the Magellan Application Server:

PR-1.8.1 Magellan will support a dedicated NT process referred to as the Magellan Data Engine that will handle all connections to one or more Magellan databases, support all required synchronization functions, and manage all versions of the Data Model for one or more Magellan databases.

PR-1.8.2 The Magellan Data Engine will support connections to Microsoft SQL Server, Versions 6.5 and 7.0, running on Windows NT 4.0 servers.

PR-1.8.3 The Magellan Data Engine will support connections to Oracle, Versions 8 and 7.3, running on Windows NT 4.0 and UNIX-based servers.

PR-1.8.4 Each Magellan Data Engine will support connections to one Magellan Database instance, containing the data required by one or more Applications within a Magellan Domain. Multiple Data Engines are required to support multiple Databases.

PR-1.8.5 **Data Model Management:** For each of the Database instances maintained by a Magellan Application Server for a particular Domain, Magellan will utilize the Data Model components of the Applications which have been deployed against that database to maintain the structure of that database. When an Application containing Data Model components is deployed, the Application Server will prepare the DDL necessary to either create or modify the structure of that database.

PR-1.8.6 The Magellan Data Engine will support the following ANSI SQL data types: VARCHAR (String), SMALLINT (Short), INTEGER (Long), DOUBLE, TIMESTAMP (Date and Time), and LONGVARBINARY (Blob)

PR-1.8.7 The Magellan Data Engine will support a special meta-data type called KEY for specifying columns which are either Primary or Foreign keys of a particular database table.

PR-1.8.8 All unique database keys will be generated for the database by the Magellan Data Engine (i.e., no database IDENTITY columns).

PR-1.8.9 Magellan will support several meta-data types, including FILE and IMAGE, which will be stored as binary data in the actual database.

PR-1.8.10 **Synchronization:** Magellan will support bi-directional synchronization of data between multiple clients and the back-end database. The model for database synchronization utilized by Magellan is captured in a separate document entitled, **Project Magellan – Synchronization Model**. Please reference that document for more detailed specifications and design. In general, however, Magellan will support an extensible model that will be used to determine both how data is distributed to clients and how client changes are ultimately applied to the back-end database.

PR-1.8.11 The Magellan Modeler will support a graphical interface for specifying the rules which will govern the data distribution methods used by the Magellan Application Server during synchronization operations.

PR-1.8.12 The Magellan Data Engine will support certain audit and status fields on all tables in the Magellan database, including fields to track Creation and Update timestamps, Creator and Updater identities, and Status fields for tracking Obsolete, Pending, and Active data rows.

PR-1.8.13 The Magellan Data Engine will support the ability to INSERT one or more rows into one or more tables in the database, including the ability to set multiple fields. All INSERTs will be made in response to CREATE calls on objects at the Magellan Foundation Interface level.

PR-1.8.14 The Magellan Data Engine will support the ability to PUBLISH one or more rows into one or more tables in the database, an operation which executes an actual CREATE, but with a Pending status and a timestamp when the data will actually become visible to other clients.

PR-1.8.15 The Magellan Data Engine will support the ability to SELECT one or more rows from one or more tables in the database, including the ability to selectively retrieve one or more fields from one or more of the tables, filter according to selective criteria, and order the results according to the clients needs. All SELECTs will be made in response to READ or QUERY calls on objects at the Magellan Foundation Interface level.

PR-1.8.16 The Magellan Data Engine will support the ability to UPDATE one or more rows in one or more tables in the database, including the ability to set multiple fields. All UPDATEs will be made in response to SAVE calls on objects at the Magellan Foundation Interface level.

PR-1.8.17 The Magellan Data Engine will support the ability to DELETE one or more rows from one or more tables in the database, according to selective criteria. All SELECTs will be made in response to DELETE calls on objects at the Magellan Foundation Interface level.

PR-1.8.18 The Magellan Data Engine will support the ability to OBSOLETE objects in one or more tables in the database, rather than an actual database DELETE. The Magellan Data Modeler will consequently support the ability to configure tables which desire this behavior.

2. Client Features

A key element of the Magellan product is its direct support for handhelds based on the Windows CE and Palm OS platforms. This support is delivered through modules referred to as the Magellan Client engines. The features listed below describe the particular capabilities of the Magellan Clients that deliver benefits to users. These capabilities provide the fundamental basis for all product definition, requirements, and project management.

2.1. General

The following requirements detail the general constraints that govern the development of the Magellan Client engines, including operating systems support, high-level module descriptions and platform-level dependencies on other software products:

PR-2.1.1 Magellan will support a Magellan Client engine for Windows CE Version 2.1 Handheld PC (H/PC) devices running MIPS and Hitachi SH3 processors. Additional processor support will be added as necessary

PR-2.1.2 Magellan will support a Magellan Client engine for Windows CE Version 3.x Handheld Pro (i.e. Jupiter class) devices running the most prevalent processors. Additional processor support will be added as necessary.

PR-2.1.3 Magellan will support a Magellan Client runtime for Palm OS, Version 3.0 devices.

PR-2.1.4 **Magellan Colonist**: Magellan will support a generic one-file starter program referred to as the Magellan Colonist that can be installed, sent via email, or downloaded to a handheld in order to handle initial deployment of the Magellan Client and User applications to the device.

PR-2.1.5 **Client Foundation**: The Magellan Client will support a master process referred to as the Client Foundation which hosts all other modules in the Magellan Client. In addition, the Client Foundation supports a set of COM or C++-based interfaces for programmatic access directly to the services (e.g., data access) required by client Application components.

PR-2.1.6 **Client Messenger**: The Magellan Client will support a process referred to as the Client Messenger dedicated to handling connectivity between the client and server-based Magellan Messenger processes or between the client and the Magellan Desktop.

PR-2.1.7 **Client Data Engine**: The Magellan Client will support a process or library referred to as the Client Data Engine dedicated to handling connections to one or more local databases, supporting data synchronization with the server, and managing the data model for one or more local databases.

PR-2.1.8 **Client Application Manager**: The Magellan Client will support a process or library referred to as the Client Application Manager dedicated to receiving and integrating new or updated Magellan applications and core modules.

PR-2.1.9 The Magellan Client for Windows CE will support applications written in either Microsoft Visual C++, Microsoft Visual Basic, or Java. "Applications" composed mainly of HTML documents will also be supported.

PR-2.1.10 The Magellan Client for Palm OS will support applications written in C or C++. The Palm version of the Foundation Services will be accessible as a standard shared code library on the handheld.

2.2. Magellan Colonist

PR-2.2.1 Magellan will support a generic one-file starter program referred to as the Magellan Colonist that can be installed, sent via email, or downloaded to a handheld from a website in order to handle initial deployment of the Magellan Client and User applications to the device.

PR-2.2.2 The Magellan Colonist program will contain a mini Client Messenger for use in connecting to the Magellan Application Server, authenticating the user, and downloading transaction files as needed. The Colonist Messenger can ONLY download. It will NOT have the ability to upload files to the server.

PR-2.2.3 The Magellan Colonist executable file will contain an embedded IP address and port number so that users do not have to enter that information to synchronize the first time. The Magellan Developer will support the ability to "encode" a generic Colonist executable with the desired information.

PR-2.2.4 The Magellan Colonist will support a simple user-interface for first-time users to enter their username and password. The password is expected to be the one that was generated at the time the user was first created in the Magellan Admin interface. The user will also be offered the opportunity to immediately change their password.

PR-2.2.5 During the first synchronization, the Magellan Colonist will connect to the Magellan Application Server, download the latest core Magellan Client files, and install the Client on the handheld. The Magellan Colonist will also download any other required configuration information which must be set in order for the handheld to start the initial download of its applications and data.

PR-2.2.6 Once the latest Magellan Client has been installed, the Magellan Colonist will support the download of the user's required applications and data. Once the user's application is installed, it is assumed that from that point on that the application itself will present the primary interfaces required for user interaction, synchronization, and configuration.

PR-2.2.7 Once one or more user applications have been installed, the Magellan Colonist will still be available to handle cases where the system has become corrupted and a re-install is required. On Windows CE, the Magellan Colonist will be accessible either as a Control Panel applet or as clickable icon in the 'Magellan' directory. On Palm OS, the Colonist will be accessible as a normal applet.

PR-2.2.8 The Magellan Colonist will support the ability for client Users to initiate a full re-deployment of their applications and data, in cases where it is clear that they local system is corrupted and should be re-installed. This capability should only be used as a last resort measure in cases where a local database has been corrupted and the user cannot contact a system administrator on the server-end to troubleshoot and potentially fix the problem for them.

2.3. Client Foundation

PR-2.3.1 The Magellan Client will support a master process referred to as the Client Foundation which hosts all other modules in the Magellan Client.

PR-2.3.2 From a system perspective, the primary role of the Client Foundation is to "orchestrate" the inner workings of the Client runtime, dispatching commands to various modules, including the Client Messenger, Data Engine, and Application Manager to execute desired operations.

PR-2.3.3 The Client Foundation will support the ability to execute a Synchronization Sequence which will utilize a Client Messenger

PR-2.3.4 The Client Foundation will support the ability to replace all modules in the core Client with newer versions which have been deployed from the server, including newer versions of the Foundation executable itself.

PR-2.3.5 From the perspective of each Magellan Application, the primary role of the Magellan Client Foundation is to present a set of services that allow the application to read and write data and synchronize with the Magellan Server.

PR-2.3.6 On Windows CE, the Client Foundation will support a set of COM-based interfaces to provide services to client applications. These are expected to be dual interfaces, supporting both custom and IDispatch (Automaton) interfaces.

PR-2.3.7 On Palm OS, the Client Foundation will support a set of C-based interfaces to provide services to client applications. These are expected to be provided in a shared code library form that matches the standard of the Palm OS platform.

PR-2.3.8 For each Magellan Application that is deployed, the Magellan Client will use that application's Data Model to present an object-oriented interface to all available local data.

PR-2.3.9 Through the Magellan Client Object model, Magellan will support an interface for client applications to programmatically initiate a synchronization / connectivity session with the Magellan Server.

2.4. Client Messenger

The following requirements detail the Connectivity features of the Magellan Client engine, outlining its support for connecting to one or more Magellan Application Servers:

PR-2.4.1 The Magellan Client will support a process referred to as the Client Messenger dedicated to handling connectivity between the client and server-based Magellan Messenger processes or between the client and the Magellan Desktop.

PR-2.4.2 The Client Messenger is considered a part of the core Magellan Client, meaning that it can be instantiated, updated, and terminated by the Client Foundation process.

PR-2.4.3 The Client Messenger will be developed according to the Requirements listed in Section 1.3 regarding general connectivity between Magellan Clients and Servers.

PR-2.4.4 The Client Messenger on Windows CE will provide the capability to make RAS connections with a Windows NT Server by interfacing with the core Win32 RAS functions that are supported by Windows CE.

PR-2.4.5 The Client Messenger on Windows CE will utilize the Winsock API on Windows CE for all TCP/IP communications to the Magellan Application Server.

PR-2.4.6 While initiating connections to the server-based Messenger, each Magellan client will provide a valid Username and Password for authentication purposes. These logon parameters will be provided to the client Messenger during its initialization by the Client Foundation Services.

PR-2.4.7 To support Client-to-Desktop connectivity on Windows CE, the Magellan Client will support a special Messenger version that will function effectively as a Windows CE platform component for ActiveSync. Synchronization files will be transferred to and from the CE-based Messenger ActiveSync component to a Magellan Desktop counterpart on the PC.

PR-2.4.8 To support Client-to-Desktop connectivity on Palm OS, the Magellan Client will provide handheld-side interfaces (as needed) in order to support the Magellan Desktop component which acts as a HotSync Manager conduit on the PC.

PR-2.4.9 The Magellan Client will NOT support the ability to intermix connectivity types , meaning that you cannot synchronize with a Magellan Desktop during one session, then directly connect to the Application Server in the next. It is possible to switch types, but the switch will require a mandatory retransact step for the client.

2.5. Client Application Manager

The following requirements detail the Application Management features of the Magellan Client, outlining its support for the deployment and integration of new and updated handheld applications:

PR-2.5.1 The Magellan Client will support a process or library referred to as the Client Application Manager dedicated to receiving and integrating new or updated Magellan applications and core modules.

PR-2.5.2 The Client Application Manager is considered a part of the core Magellan Client, meaning that it can be instantiated, updated, and terminated by the Client Foundation process.

PR-2.5.3 The Client Application Manager will be developed according to the Requirements listed in Section 1.7 regarding general Application Management for Magellan Clients and Servers.

PR-2.5.4 The Client Application Manager will maintain a catalog of each Magellan Application which has been deployed to the handheld, including versions and locations of its Program, Extension, Library, and Data Model components.

PR-2.5.5 The Client Application Manager will support the ability to "receive" and catalog one or more Transaction Files from a synchronization sequence that manifest a Deployment Set—meaning that they contain new or updated versions of Application Components, including Data Models. The Application will optionally keep multiple versions of successive Deployment Sets in order to rollback applications to their previous state.

PR-2.5.6 The Client Application Manager will support the ability to integrate a new version of an Application by unpacking a Deployment Set of transaction files, initiating the shutdown of the current version of a client application, publishing a new Data Model, replacing the component files, and restarting the client application.

PR-2.5.7 The Client Application Manager will support a safety feature which prevents a new or updated version of an Application from being applied until all required Transaction Files have been received and successfully unpacked..

PR-2.5.8 The Client Application Manager will support the ability to publish a new Data Model by unpacking it from a Transaction File and "sending" it to the current Data Engine via the Client Foundation.

PR-2.5.9 The Client Application Manager will support a safety feature which prevents a new Data Model from being published before the Application Manager can verify that it will also be able to install the application components which are dependent on it.

2.6. Client Data Engine

The following requirements detail the Data Management features of the Magellan Client:

PR-2.6.1 The Magellan Client will support a process or library referred to as the Client Data Engine dedicated to handling connections to one or more local databases, supporting data synchronization with the server, and managing the data model for one or more local databases.

PR-2.6.2 The Client Data Engine is considered a part of the core Magellan Client, meaning that it can be instantiated, updated, and terminated by the Client Foundation process.

PR-2.6.3 The Client Data Engine will be developed according to the Requirements listed in Section 1.8 regarding general Data Management for Magellan Clients and Servers.

PR-2.6.4 The Client Data Engine will support local storage of data on Windows CE clients using the native Object Store, accessing this store using ADO (ActiveX Data Objects) for Windows CE. Support for additional local databases (e.g., Sybase, Oracle Lite, etc.) will be added as necessary.

PR-2.6.5 The Client Data Engine will support local storage of data on Palm OS clients using the local Palm database, accessing this store through native operating system interfaces. Support for additional local databases (e.g., Sybase, Oracle Lite, etc.) will be added as necessary.

PR-2.6.6 The Client Data Engine will support an interface that allows the Client Application Manager to publish new or updated versions of an Application's Data Model. The Client Data Model will use that application's Data Model to create and maintain the structure of the local database. As changes to that Data Model are deployed, the Magellan Client will alter the local database to comply.

3. Desktop Features

For customers that do not have the infrastructure to support direct handheld-to-server connections, Magellan will support an intermediary module referred to as the Magellan Desktop that can facilitate the link. The features listed below describe the particular capabilities of the Magellan Desktop that deliver benefits to users. These capabilities provide the fundamental basis for all product definition, requirements, and project management.

3.1. General

The following requirements detail the general constraints that govern the development of the Magellan Desktop, including operating systems support, high-level module descriptions and platform-level dependencies on other software products:

PR-3.1.1 The Magellan Desktop will run on the latest versions of Windows NT 4.0, Windows 95, and Windows 98.

PR-3.1.2 Magellan will support the capability for Magellan Clients to synchronize with the Application Server through the Desktop when the handheld is docked.

PR-3.1.3 **Desktop Foundation:** The Magellan Desktop will support a process referred to as the Desktop Foundation dedicated to facilitating connections between Magellan Clients and the Magellan Application Server. The Magellan Desktop accomplishes this by both acting as a staging area for incoming and outgoing data and as a host for other modules required to communicate with client and server processes.

PR-3.1.4 **Desktop Messenger:** The Magellan Desktop will support a process referred to as the Desktop Messenger dedicated to handling connectivity between the PC and server-based Magellan Messenger processes.. The Desktop Messenger module can be found in both Magellan Desktops and Magellan Developer products.

3.2. Desktop Foundation

The following requirements detail the features of the Desktop Foundation, outlining its support for facilitating connections between multiple Magellan Clients and the central server:

PR-3.2.1 The Magellan Desktop will support a process referred to as the Desktop Foundation dedicated to facilitating connections between Magellan Clients and the Magellan Application Server.

PR-3.2.2 The Desktop Foundation supports the capability to cache or stage sets of pending transaction data for one or more client handhelds. This capability should work as follows: The Magellan Client communicates with the Desktop Foundation, through a docking station, using one of the supported Magellan Desktop Providers. This connection will allow the upload and download of Transaction Files between the PC and the handheld. The Desktop Foundation then communicates with the Magellan Application Server, using its own Desktop Messenger, for the purpose of uploading and downloading Transaction Files for each of the users currently configured to use that Desktop. The Desktop can then communicate again with one or more handhelds or synchronize again with the server

PR-3.2.3 **Magellan ActiveSync Provider:** The Magellan Desktop will support an in-process component referred to as the Magellan ActiveSync Provider which will facilitate file transfers between the PC and the Windows CE handheld using the Windows CE Services interfaces.

PR-3.2.4 **Magellan HotSync Provider:** The Magellan Desktop will support an in-process component referred to as the Magellan HotSync Provider which will facilitate file transfers between the PC and the Palm OS handheld using the HotSync Manager Conduit interfaces.

PR-3.2.5 The Desktop Foundation will support a scheduling capability in order to facilitate automated synchronization sessions between the Desktop PC and the Magellan Application Server.

PR-3.2.6 The Desktop Foundation will provide a basic user-interface for configuring the particular client Users that will connect to the Desktop. The configuration will include the logon parameters for each client, their client type (e.g., Windows CE, Palm OS), and the connection parameters for the desired server (e.g., IP Address, Port Number, etc…)

PR-3.2.7 The Desktop Foundation will support client Users that wish to connect to different Magellan Application Servers, as long as the Desktop PC can make a valid TCP/IP connection to all the servers required by all Users. The Desktop Foundation will NOT support more than one Application Server per User.

3.3. Desktop Messenger

The following requirements detail the Connectivity features of the Magellan Desktop, outlining its support for connecting between the Desktop PC and the central server:

PR-3.3.1 The Magellan Desktop will support a process referred to as the Desktop Messenger dedicated to handling connectivity between the PC and server-based Magellan Messenger processes.

PR-3.3.2 The Desktop Messenger will be developed according to the Requirements listed in Section 1.3 regarding general connectivity between Magellan Clients and Servers.

PR-3.3.3 The Desktop Messenger will support both a direct-to-network configuration which assumes a full-time live TCP/IP capability and a Remote Access configuration where the PC will need to dialup a RAS server to access the Magellan Application Server. In the Remote Access configuration, the Desktop will provide the capability to make RAS connections with a Windows NT Server by interfacing with the core Win32 RAS functions that are supported.

PR-3.3.4 The Desktop Messenger will utilize the Winsock API for all TCP/IP communications to the Magellan Application Server.

PR-3.3.5 While initiating connections to the server-based Messenger, the Desktop Messenger will need to provide a valid Username and Password for all Users which are currently using the Desktop for connection purposes. These logon parameters will be provided to the Desktop Foundation interface during the configuration of each client.

4. Developer Features

For the Application Developers that are counting on the Magellan product to provide valuable services for their code to call upon, the Magellan Developer is a crucial module in the overall system. The features listed below describe the particular capabilities of the Magellan Developer that deliver benefits to users. These capabilities provide the fundamental basis for all product definition, requirements, and project management.

4.1. General

The following requirements detail the general constraints that govern the development of the Magellan Developer, including operating systems support, high-level module descriptions and platform-level dependencies on other software products:

PR-4.1.1 The Magellan Developer module will initially support development on Windows NT Workstation and Server, Version 4.0, with SP4.

PR-4.1.2 Magellan will support Windows 2000 (NT 5.0) Server and Advanced Server within 90 days of their support in a general release of Microsoft Visual Studio.

PR-4.1.3 For Windows CE clients, the Magellan Developer interface will be integrated into the Microsoft Visual Studio development environment through standard Developer Studio Add-In interfaces. Development will be supported only on the latest version of Windows NT, Version 4.0

PR-4.1.4 For Palm OS clients, the Magellan Developer interface will be integrated into the Windows 95 and NT versions of Metrowerks CodeWarrior environment for Palm.

PR-4.1.5 **Application Manager**: The Magellan Developer will support an interface referred to as the Application Manager dedicated to providing an interface to the Magellan Development Server.

PR-4.1.6 **Magellan Data Modeler**: The Magellan Developer will support an interface referred to as the Magellan Data Modeler. Developers will use the Data Modeler to represent all required application data as objects, mapping each to physical database tables, columns, and joins. In addition, Data Distribution Relationships are also created to facilitate synchronization of data to Users, Groups, and Roles.

PR-4.1.7 **Magellan Framework**: The Magellan Developer will support a set of C++ class libraries and templates referred to as the Magellan Framework dedicated to increasing the efficiency of developers that are writing code to access objects in the Magellan Foundation.

4.2. Application Manager

The following requirements detail the Application Management features of the Magellan Developer product, outlining its support for the development, management, and deployment of applications to handheld devices:

PR-4.2.1 The Magellan Developer will support an interface referred to as the Application Manager dedicated to providing an interface to the Magellan Development Server.

PR-4.2.2 The Developer Application Manager will be developed according to the Requirements listed in Section 1.7 regarding general Application Management for Magellan Clients and Servers.

PR-4.2.3 A key function of the Magellan Developer product will be to support connections to the server-based Application Manager which controls access to the Magellan Component Repository, a dedicated database for storing instances of supported Magellan Component Classes.

PR-4.2.4 Magellan will support an interface in the Magellan Developer that allows programmers to add, remove, and maintain instances of the various Magellan Component Classes

PR-4.2.5 Magellan will support a full version control capability in the Magellan Developer with the capability to checkout, check-in, and label different versions of individual application components.

4.3. Magellan Data Modeler

PR-4.3.1 Magellan will support an extension to the Magellan Developer product called the Magellan Modeler for maintaining Data Model components.

PR-4.3.2 The Magellan Modeler will provide a graphical interface for maintaining the set of Tables which are contained in an Magellan Data Model component, each encapsulating properties such as Name. For each Magellan Table, the Magellan Modeler will support the maintenance of one or more Columns, each specifying properties such as Name, Data Type, and Index.

PR-4.3.3 The Magellan Modeler will support a graphical interface for specifying the desired connections between the various tables in the Data Model, including joins, constraints, and ownership attributes which will be used by the Magellan Application Server during synchronization operations.

PR-4.3.4 As developers use the Magellan Data Modeler to detail the database structure, the Magellan Data Modeler will at the same time create a complementary Data Access Object Model that will "overlay" the Data Model in order to provide an interface for Magellan Client applications to access the data.

PR-4.3.5 A separate document entitled, **Project Magellan – Data Access Model** covers more detail on the design of Data Access Object Model. In general, however, Magellan will support the concept that each table in the database will have a matching Object Class, with properties that map to each of the fields in the table. Calls to methods on objects of these classes will then be translated into DML commands which are executed against the connected database tables. Assessors to properties on objects of these classes will then be accessing data from the linked field. The Magellan Data Modeler will support a graphical interface for reviewing Object Classes and optionally changing the names of object classes and properties.

PR-4.3.6 The Magellan Modeler will support a graphical interface for specifying the rules which will govern the data distribution methods used by the Magellan Application Server during synchronization operations. More detail on the requirements for this area can be found in a separate document entitled, **Project Magellan – Synchronization Model.**

4.4. Magellan Framework

PR-4.4.1 Magellan will support an MFC-based class library and Active Template Library (ATL)-based templates for assisting the development of Magellan Applications for Windows CE handhelds

PR-4.4.2 Magellan will provide C++ class library support for assisting the development of Magellan Applications for Palm OS handhelds

5. Miscellaneous Requirements

5.1. Internationalization

Localization is the process of adapting software products to the needs of a local market. Internationalization is the process of developing software products independent from cultural peculiarities, language or other specific attributes of a market. All Magellan modules will be developed using standard internationalization-savvy methods and processes so that localization is possible for a wide range of markets, including Far East multi-byte languages.

5.2. Performance

Performance encompasses both the speed with which the application performs a task, especially as perceived by the user, and the level of availability of the application. All Magellan modules will be designed and developed in such a way that performance considerations are accounted for at every milestone of the project.

5.3. Support and Maintenance

Requirements TBD

5.4. Installation

In general, the Magellan product will be available via CD-ROM with options to selectively install components or upgrades over the Internet. Specifics regarding logos, product name, and other product packaging considerations will be determined by Product Management.

5.5. Licensing

Magellan will offer separate licenses for each component, including Magellan Clients, Application Server, Integration Server, and Developer seat. In addition, the Application and Integration Servers may be offered in "Production" and "Development" licenses at different price points. The licensing model also contemplates licenses restricting Magellan for use only with specific back-end applications such as, for example, third-party OEM applications.

@hand

Project Magellan

Product Requirements Document
Version 1.03

IMPORTANT

This document contains information proprietary to @hand Corporation and as such this document is generally provided to outside parties only under Non-Disclosure Agreement and may only be used to clarify particular product and/or strategic intents currently proposed or under discussion by @hand Corporation. No other uses are permitted. Furthermore, the information in this proposal remains the property of @hand and may not be reproduced or redistributed without the express written approval of @hand Corporation.

Table of Contents

Revisions

Version	Date	Description
0.9	September 1998	Working version created to explore initial product concepts
1.0	October 26, 1998	First Official DRAFT
1.01	November 07, 1998	Second Official DRAFT
1.02	December 01, 1998	Updated format. Aligned with first version of Project Plan.
1.03	December 11, 1998	Incorporated revisions. Synchronized with Architecture documents. Significant expansion of detail in Application Server and Client features.